Game Theoretic Analysis and Agent-Based Simulation of Electricity Markets

by

Teruo Ono

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2005
[June 2005]

Author ....
Department of Electrical Engineering and Computer Science
May 6, 2005

Certified by ....
George C. Verghese
Professor
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Game Theoretic Analysis and Agent-Based Simulation of Electricity Markets

by

Teruo Ono

Submitted to the Department of Electrical Engineering and Computer Science
on May 6, 2005, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

In power system analysis, uncertainties in the supplier side are often difficult to estimate and have a substantial impact on the result of the analysis. This thesis includes preliminary work to approach the difficulties. In the first part, a specific electricity auction mechanism based on a Japanese power market is investigated analytically from several game theoretic viewpoints. In the second part, electricity auctions are simulated using agent-based modeling.

Thesis Supervisor: George C. Verghese
Title: Professor

Acknowledgments

First of all, I would like to thank to my research advisor Prof. George C. Verghese. This thesis cannot be done without his suggestion, guidance and encouragement. I really appreciate his dedicating time for me while dealing with important works as a professor and an educational officer.

I also thank to my officemates Tushar Parlikar, Victor Preciado, Carlos Renjifo and all LEES members for their support. My life at MIT has been incredibly enriched and wonderful. My wife also has been always understanding and supportive to my work as a graduate student.

Finally, I gratefully acknowledges fellowship support from the Tokyo Electric Power Company.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In a large number of countries and regions, auctions have been adopted or will be adopted as a means of transaction in electricity markets. To understand how these auction systems work is crucial not only for participants of the auction but also for power system operators. Auctions affect the income and the expenditure of the participants as well as network security.

For participants of the auction, bidding strategies for the auction are of interest, and these have been explored under various settings. The optimal bidding strategy can be derived by backward dynamic programming, with its target function set to cost, when the market clearing price is known from historical data [1]. Also, when the residual demand curve is known from historical data, the optimal bidding strategy is derived by dynamic programming [1][2][3]. As a result, estimating the market clearing price or the residual demand curve precisely is imperative work for participants of the auction to maximize their profit.

For electricity market designers, this market clearing price is of interest. If the electricity auctions were held only once, characterizing the market clearing price as a Nash equilibrium would be rather simple. Since the electricity auctions are held repeatedly everyday, however, there are a myriad of Nash equilibria in this game. As is known from repeated Prisoner's Dilemma experiments, players often cooperate in repeated games, which makes the expected market clearing price higher. These experiments show that a player cooperates first and watches how the other players behave expecting that the other players don't use backward induction to maximize their profit either. In this way, the players can accomplish Pareto efficient operation, which is also called tacit collusion in this case. However, tacit collusion is not stable and breaks down to a non-Pareto optimal Nash equilibrium during the repeated games. After the breakdown, the strategy in which the players both defect until the last game is a Nash equilibrium, which is called trigger strategy, but in the experiments, the players often tacitly renegotiate to accomplish Pareto efficiency [4]. These experiments suggest the difficulty of predicting the market clearing price in repeated auctions.

In order to deal with this complexity of repeated games, agent-based simulation has been utilized recently. Using agent-based simulation, under the setting in which the agents with the highest 0.5% payoff reproduce themselves, replacing the agents with the lowest 0.5% payoff, high-cooperation agents are better off when the incentive to defect is low. In contrast, low-cooperation agents are better off when the incentive to defect is high [4]. This example suggests the difficulty of estimating how the auction system works. Extending this game, electricity auctions in England and Wales [5][6] and in the United States have been analyzed using agent-based simulation in order to investigate the markets from the aspect of the market clearing price, collusion, and the exercise of the market power.

In this thesis, repeated electricity auctions are analyzed from the game theoretic viewpoint. The electricity auction system used in a particular Japanese power market is assumed, and its simplest form is modeled to help better understand how the auction system works. In Japan, auctions have been adopted for electric power transactions, and auction-based transactions are expected to continue to grow due to the increase of generating power from independent Power Producers and Suppliers (PPSs). In order to support the transaction, the Japan Electric Power Exchange (JEPX) was founded last year, and will start its operation from April 2005.

At first, this repeated game is investigated analytically using different approaches, and later, the game is simulated using agent-based modeling. In Chapter 2, classical game theoretic techniques are used to analyze the non-repeated auction. The assumed auction system and the background are also described in this chapter. In Chapter 3, the strategies in the repeated auction are modeled via a Markov chain, and the dynamics of the game are explored. The dynamics of the game are also analyzed using replicator equations in Chapter 4. Finally, the repeated auction is investigated by agent-based simulation in Chapter 6. Learning algorithms for the agents are carefully considered to verify the validity of the simulation.

Chapter 2

Non-Repeated Game Analysis

2.1 Auction Model

In this chapter, the auction is restricted to be held only once. Players have no information about the other players.

> **Assumed Model**
>
> A company, PPS0, announces its intent to buy d MWh of electricity with its maximum acceptable price P at the auction, and two companies, PPS1 and PPS2, bid to sell d MWh of their electricity to PPS0. Here, P is the market reserve price, and a bid price higher than or equal to P is not acceptable. Both PPS1 and PPS2 have sufficient generating capacity to supply d to PPS0. Their bid prices to PPS0 are p_1 and p_2, respectively, and their respective costs to supply d to PPS0 are c_1 and c_2. Here, $c_1 < p_1 < P$, $c_2 < p_2 < P$, and assume without loss of generality that $c_1 < c_2$.

This model is the simplest form of the electricity auction system used in a particular Japanese power market, where PPSs and large utilities participate in the auction. The Japanese auction system consists of two rounds of auctions. At the first round of the auction, first, some companies, sellers, announce their intent to sell their electricity. Sellers have surplus electricity in addition to the demand they have to supply, and selling their surplus electricity at the auction is more economical for them than stopping some generators or reducing their generating output. Second, other companies, buyers, bid to buy electricity. Some buyers bid to buy electricity because they don't have sufficient electricity to supply their demand. The other buyers have sufficient electricity to supply their demand, but buying electricity at the auction is more economical for them than producing electricity by themselves. The seller announcements are made with the amount for sale and the minimum acceptable price, and the buyer bid prices must be higher than this minimum acceptable price. Last, sellers choose buyers according to buyer bid prices, from the higher bid price to the lower bid price, until the total amount of electricity reaches the amount they specified in their announcements. The first round of the auction closes at this point. Sellers are responsible for producing electricity for the buyers they chose, and buyers are responsible for paying their bid prices.

As soon as the first round of the auction closes, the second round of the auction starts. At the second round of the auction, first, some companies, buyers, announce their intent to buy electricity. Some buyers announce to buy electricity because they don't have sufficient electricity even after the first round of the auction, and the other buyers announce because buying electricity at the auction is more economical for them than producing electricity by themselves. Second, other companies, sellers, bid to sell their electricity. These sellers have surplus electricity even after the first round of the auction. One reason is that their minimum acceptable prices were too high at the first round, and they didn't receive enough buyer bids to sell out surplus electricity. Another reason is that they didn't sell out surplus electricity on purpose expecting that they could sell their electricity at a higher price at the second round. For these sellers, selling their surplus electricity at the auction is more economical for them than stopping some generators or reducing their generating output. The buyer announcements are made with the amount to buy and the maximum acceptable price, and the seller bid prices must be lower than this maximum acceptable price. Last, buyers choose sellers according to seller bid prices, from the lower bid price to the higher bid price, until the total amount of electricity reaches the amount they specified in their announcement. The second round of the auction closes at this point. Sellers are responsible for producing electricity for the buyers they chose, and buyers are responsible for paying their bid prices.

The assumed model in this thesis represents the second round of the auction. In order to understand how the auction works, we assume only one company announces to buy electricity. Also, the number of companies that bid to this announcement is limited to two.

2.2 Pure-Strategy Nash Equilibrium

Pure-strategy Nash equilibrium is the equilibrium under which each player chooses only one strategy. In this model, the bid price of each company is its strategy, and each company chooses one bid price with 100% probability for the single auction. In contrast, under the mixed-strategy Nash equilibrium, each player can have multiple strategies, and each strategy is chosen with a probability specified in the equilibrium.

For our model, we can argue as follows that there is a unique pure-strategy Nash equilibrium, in which PPS1 bids with the price $p_1 = c_2$ and wins the auction. The proof is given as follows. If PPS1 bids with $p_1 > c_2$, PPS2 can undercut PPS1's bid by $p_2 < p_1$. However, PPS1 can always undercut this PPS2's bid by $p_1 < p_2$ because p_2 cannot be smaller than or equal to c_1. Moreover, among the PPS1's possible bids between c_1 and c_2, which assures PPS1's win, the bid $p_1 = c_2$ brings the best profit to PPS1. Thus, bidding with $p_1 = c_2$ is the best strategy for PPS1 regardless of PPS2's bid.

2.3 Iterated Elimination of Dominated Strategies

A Nash equilibrium may also be found by the iterated elimination of dominated strategies. The following example highlights the process by which Nash equilibrium is derived, and suggests the nature of Nash equilibrium in the assumed model in this thesis.

Example

In the assumed model in this thesis, let $c_1 = 5.0$, $c_2 = 7.0$, and $P = 12.0$, and also suppose that the bid price is discrete with an interval of 1.0 between bids. Thus, the acceptable bid price for PPS1 is $c_1 + 1.0 \times i$ (i = 1,2,3,4,5,6), and the acceptable bid price for PPS2 is $c_2 + 1.0 \times i$ (i = 1,2,3,4).

If $p_1 = p_2$, PPS1 and PPS2 both sell $\frac{d}{2}$ MWh of electricity to PPS0 at the price $\frac{p_1}{2}$. Assume that the cost to sell $\frac{d}{2}$ MWh of electricity is $\frac{c_1}{2}$ and $\frac{c_2}{2}$ for PPS1 and PPS2, respectively. Though we can assign the monetary unit for p_1, p_2, c_1, and c_2, we define that they have no unit except Chapter 6. Note that p_1, p_2, c_1, and c_2 are not given per MWh, and that is the reason that we assume half price and half cost for supplying half amount of electricity.

Also, note that these conditions are given in order to make the number of strategies countable, and they don't limit the generality of the assumed model.

In this case, PPS1's bid price is one of 6.0, 7.0, 8.0, 9.0, 10.0, or 11.0. PPS2's bid price is one of 8.0, 9.0, 10.0, or 11.0, so PPS1 can always win when $p_1 = 6.0$ or 7.0. Since PPS1's strategy $p_1 = 6.0$ is obviously strictly dominated by the strategy $p_1 = 7.0$, assume that PPS1's bid price is one of 7.0, 8.0, 9.0, 10.0, or 11.0.

The iterated elimination of dominated strategies is often performed using the payoff matrix. In this case, the payoff for PPS1 and PPS2 is the profit from the auction. Let u_1 and u_2 be the payoff for PPS1 and PPS2. When PPS1 wins the auction, $u_1 = p_1 - c_1$ and $u_2 = 0$. When PPS2 wins the auction, $u_1 = 0$ and $u_2 = p_2 - c_2$. When they tie, $u_1 = \frac{p_1 - c_1}{2}$ and $u_2 = \frac{p_2 - c_2}{2}$. The objective of both PPS1 and PPS2 is to maximize this payoff from the auction.

Table 2.1 is the payoff matrix of this case. The payoff (u_1, u_2) is shown in the table. When PPS1 and PPS2 choose their strategies, if no improvement of u_1 can be made by changing p_1 without lowering u_2, and if no improvement of u_2 can be

		PPS2's Strategy (p_2)			
		8.0	9.0	10.0	11.0
PPS1's Strategy (p_1)	7.0	(2.0, 0.0)	(2.0, 0.0)	(2.0, 0.0)	(2.0, 0.0)
	8.0	(1.5, 0.5)	(3.0, 0.0)	(3.0, 0.0)	(3.0, 0.0)
	9.0	(0.0, 1.0)	(2.0, 1.0)	(4.0, 0.0)	(4.0, 0.0)
	10.0	(0.0, 1.0)	(0.0, 2.0)	(2.5, 1.5)	(5.0, 0.0)
	11.0	(0.0, 1.0)	(0.0, 2.0)	(0.0, 3.0)	(3.0, 2.0)

Table 2.1: Payoff matrix.

made by changing p_2 without lowering u_1, the set of their strategies is called Pareto efficient. From Table 2.1, we can see that (p_1, p_2) = (10.0, 11.0), (11.0, 10.0), (11.0, 11.0) are Pareto efficient strategies.

Now, the definition of the dominated strategy is given as follows. Let $u_1(p_1, p_2)$ denote the payoff PPS1 receives when PPS1 and PPS2 choose the strategy p_1 and p_2. When we have following inequality:

$$u_1(i,k) > u_1(j,k),\ for\ {}^{\forall}k \in p_2\ ,$$

we say PPS1's strategy $p_1 = j$ is strictly dominated by its strategy $p_1 = i$. Similarly, when we have following inequality:

$$u_1(i,k) \geq u_1(j,k),\ for\ {}^{\forall}k \in p_2\ ,$$

we say PPS1's strategy $p_1 = j$ is weakly dominated by its strategy $p_1 = i$.

In the payoff matrix, no strategy is strictly dominated by other strategies. However, PPS1's strategy $p_1 = 11.0$ and PPS2's strategy $p_2 = 11.0$ are weakly dominated by their other strategies. Since the iterated elimination of weakly dominated strategies results in different Nash equilibria according to the order of the iterated eliminations, careful attention has to be paid to the order of the iterated eliminations.

First, suppose $p_1 = 11.0$ is eliminated before $p_2 = 11.0$ is eliminated. After $p_1 = 11.0$ is eliminated, the payoff matrix is updated as shown in Table 2.2.

PPS2's Strategy (p_2)

PPS1's Strategy (p_1)		8.0	9.0	10.0	11.0
	7.0	(2.0, 0.0)	(2.0, 0.0)	(2.0, 0.0)	(2.0, 0.0)
	8.0	(1.5, 0.5)	(3.0, 0.0)	(3.0, 0.0)	(3.0, 0.0)
	9.0	(0.0, 1.0)	(2.0, 1.0)	(4.0, 0.0)	(4.0, 0.0)
	10.0	(0.0, 1.0)	(0.0, 2.0)	(2.5, 1.5)	(5.0, 0.0)

Table 2.2: Payoff matrix after one round of elimination.

In Table 2.2, no strategy of PPS1 is strongly or weakly dominated by its other strategies, but PPS2's strategies $p_2 = 10.0$ and 11.0 are weakly dominated by its other strategies. Thus, the column of $p_2 = 10.0$ or $p_2 = 11.0$ can be deleted now.

The tree of this iterated elimination is drawn in Figure 2-1. In Figure 2-1, different nodes in one rectangle represent the same resulting payoff matrix from different orders of eliminations. From the elimination tree, we can see that after the elimination of three strategies from Table 2.1, different orders of eliminations result in only two kinds of payoff matrices, shown in Table 2.3 and Table 2.4.

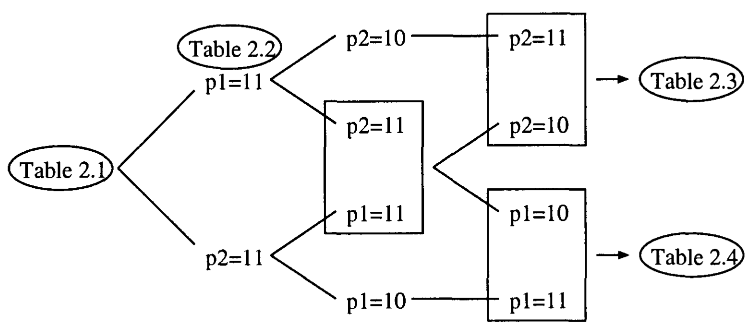

Figure 2-1: Elimination tree.

PPS2's Strategy (p_2)

PPS1's Strategy (p_1)		8.0	9.0
	7.0	(2.0, 0.0)	(2.0, 0.0)
	8.0	(1.5, 0.5)	(3.0, 0.0)
	9.0	(0.0, 1.0)	(2.0, 1.0)
	10.0	(0.0, 1.0)	(0.0, 2.0)

Table 2.3: Payoff matrix after one order of elimination.

		PPS2's Strategy (p_2) 8.0	9.0	10.0
PPS1's	7.0	(2.0, 0.0)	(2.0, 0.0)	(2.0, 0.0)
Strategy	8.0	(1.5, 0.5)	(3.0, 0.0)	(3.0, 0.0)
(p_1)	9.0	(0.0, 1.0)	(2.0, 1.0)	(4.0, 0.0)

Table 2.4: Payoff matrix after another order of elimination.

Now, from Table 2.3 and Table 2.4, we can continue to eliminate strongly or weakly dominated strategies. The result of the elimination is shown in Figure 2-2.

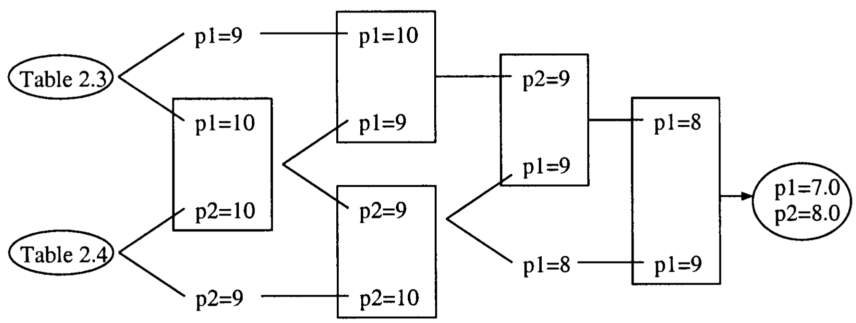

Figure 2-2: Elimination tree after Figure 2-1.

Figure 2-2 shows that the iterated elimination of dominated strategies yields (p_1, p_2) = (7.0, 8.0) as the only surviving strategy, regardless of the order of eliminations. This means that (p_1, p_2) = (7.0, 8.0) is the only pure-strategy Nash equilibrium for this case, and this result corresponds with the equilibrium stated in the previous section.

2.4 Mixed-Strategy Nash Equilibrium

Nash showed that any n-person game with a finite number of pure strategies, but with mixed strategies allowed, has at least one Nash equilibrium [7]. We have already found a pure-strategy Nash equilibrium of the non-repeated auction in our assumed model, but when we consider mixed-strategies, we might be able to find other Nash equilibria of the same non-repeated auction. The Fundamental Theorem of mixed-strategy Nash equilibrium is often used, in order to find a mixed-strategy Nash equilibrium from a payoff matrix. The Fundamental Theorem can be explained as follows.

In our assumed model, let x_i be the probability that PPS1 uses the i-th strategy. Then, when PPS1 has n strategies, the column vector $x = [x_1, x_2, ..., x_n]^T$ ($\sum_{i=1}^{n} x_i = 1$) specifies the mixed-strategy of PPS1. Also, let $y = [y_1, y_2, ..., y_m]^T$ ($\sum_{i=1}^{m} y_i = 1$) denote the mixed-strategy for PPS2. Now, we define the payoff matrix for PPS1 as:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nm} \end{pmatrix}$$

where a_{ij} is the payoff for PPS1 when it uses the i-th strategy and PPS2 uses the j-th strategy. We also define the payoff matrix for PPS2:

$$B = \begin{pmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & & \vdots \\ b_{m1} & \cdots & b_{mn} \end{pmatrix}$$

where b_{ij} is the payoff for PPS2 when it uses the i-th strategy and PPS1 uses the j-th strategy.

Next, let π_{Ai} be the expected payoff for PPS1 when it uses the i-th strategy. The payoff π_{Ai} is given as:

$$\pi_{Ai} = a_{i1}y_1 + a_{i2}y_2 + \cdots + a_{im}y_m$$

Thus, the column vector $\pi_A = [\pi_{A1}, \pi_{A2}, ..., \pi_{An}]^T$, which specifies the payoff for each strategy of PPS1, is calculated as:

$$\pi_A = Ay$$

Let $\hat{\pi}_A$ be the expected payoff when PPS1 uses the mixed-strategy x. The expected payoff $\hat{\pi}_A$ is

$$\hat{\pi}_A = x^T \pi_A = x^T Ay$$

In the same way, the payoff for each strategy of PPS2 is

$$\pi_B = Bx$$

and the expected payoff is

$$\hat{\pi}_B = y^T \pi_B = y^T Bx$$

Under this formulation, the Fundamental Theorem states that (x^*, y^*) is a mixed-strategy Nash equilibrium if and only if:

$$\begin{cases} \pi_{Ai} = \hat{\pi}_A & \Leftrightarrow \quad [Ay^*]_i = x^{*T} Ay^* \quad , \textit{ for } {}^{\forall} i \textit{ s.t. } x_i^* > 0 \\ \pi_{Ai} \leq \hat{\pi}_A & \Leftrightarrow \quad [Ay^*]_i \leq x^{*T} Ay^* \quad , \textit{ for } {}^{\forall} i \textit{ s.t. } x_i^* = 0 \\ \pi_{Bi} = \hat{\pi}_B & \Leftrightarrow \quad [Bx^*]_i = y^{*T} Bx^* \quad , \textit{ for } {}^{\forall} i \textit{ s.t. } y_i^* > 0 \\ \pi_{Bi} \leq \hat{\pi}_B & \Leftrightarrow \quad [Bx^*]_i \leq y^{*T} Bx^* \quad , \textit{ for } {}^{\forall} i \textit{ s.t. } y_i^* = 0 \end{cases}$$

where $[Ay^*]_i$ indicates the i-th row of the column vector Ay^*. It implies that each strategy of PPS1 has to yield the same payoff $\hat{\pi}_A$ for itself and each strategy of PPS2 has to yield the same payoff $\hat{\pi}_B$ for itself, in order for (x^*, y^*) to be a completely mixed-strategy Nash equilibrium. For, if PPS1 obtains a larger payoff by playing i-th strategy than j-th strategy, that is $\pi_{Ai} > \pi_{Aj}$, it can obtain a larger average payoff by playing the i-th strategy with a probability $x_i + x_j$ and not playing the j-th strategy.

Applying the Fundamental Theorem to our assumed model, in order for (x^*, y^*) to be a completely mixed-strategy it has to satisfy the following equations, derived from the payoff matrix shown in Table 2.1:

$$\left\{\begin{array}{lllllllllllll}
\hat{\pi}_A & = & \pi_{A1} & = & 2.0y_1 & + & 2.0y_2 & + & 2.0y_3 & + & 2.0y_4 & = & 2.0 \\
\hat{\pi}_A & = & \pi_{A2} & = & 1.5y_1 & + & 3.0y_2 & + & 3.0y_3 & + & 3.0y_4 & & \\
\hat{\pi}_A & = & \pi_{A3} & = & & & 2.0y_2 & + & 4.0y_3 & + & 4.0y_4 & & \\
\hat{\pi}_A & = & \pi_{A4} & = & & & & & 2.5y_3 & + & 5.0y_4 & & \\
\hat{\pi}_A & = & \pi_{A5} & = & & & & & & & 3.0y_4 & & \\
\hat{\pi}_B & = & \pi_{B1} & = & 0.5x_2 & + & 1.0x_3 & + & 1.0x_4 & + & 1.0x_5 & & \\
\hat{\pi}_B & = & \pi_{B2} & = & & & 1.0x_3 & + & 2.0x_4 & + & 2.0x_5 & & \\
\hat{\pi}_B & = & \pi_{B3} & = & & & & & 1.5x_4 & + & 3.0x_5 & & \\
\hat{\pi}_B & = & \pi_{B4} & = & & & & & & & 2.0x_5 & &
\end{array}\right.$$

Since $\pi_{A1} = 2.0$, we can see that $\hat{\pi}_A$ is equal to 2.0 if $x_1 > 0$, and $\hat{\pi}_A$ is larger than or equal to 2.0 otherwise. Now, we will show $x_5 = 0$ by contradiction. Suppose $x_5 \neq 0$. Then, $\pi_{A5} = \hat{\pi}_A \geq 2.0$ from the Fundamental Theorem. It requires $y_4 \geq \frac{2}{3}$, but $\pi_{A4} > \pi_{A5}$ for ${}^{\forall}y_3$ when $y_4 > 0$. Since $\hat{\pi}_A$ has to be larger than or equal to π_{A4} under any condition, $\hat{\pi}_A > \pi_{A5}$. This contradicts the requirement $\pi_{A5} = \hat{\pi}_A$ from the Fundamental Theorem. Thus, $x_5 = 0$. We see that a completely mixed-strategy doesn't exist, but a (not-completely) mixed-strategy may still exist.

When $x_5 = 0$, we have $\pi_{B4} = 0.0$. We need to consider two cases.

i) $\hat{\pi}_B = \pi_{B4} = 0.0$

In this case, $\pi_{B1} = \pi_{B2} = \pi_{B3} = \pi_{B4} = 0.0$, which requires $x_2 = x_3 = x_4 = x_5 = 0$. Thus, $x = [1, 0, 0, 0, 0]$ is the only Nash equilibrium for PPS1 including mixed-strategies. Under this strategy of PPS1, the payoff of PPS1 $\hat{\pi}_A = \pi_{A1} = 2.0$, and the payoff of PPS2 $\hat{\pi}_B = \pi_{Bi} = 0.0$ for ${}^{\forall}i$ for any mixed-strategy of PPS2. Therefore, any mixed-strategy of PPS2 which satisfies

$$\hat{\pi}_A = \pi_{A1} \geq \pi_{Ai} \Leftrightarrow [Ay^*]_1 \geq [Ay^*]_i \ , for\ i = 2, 3, 4, 5$$

can be regarded as a mixed-strategy Nash equilibrium for PPS2.

ii) $\hat{\pi}_B > \pi_{B4} = 0.0$

In this case, we have $y_4 = 0$ from the Fundamental Theorem. By setting $x_5 = 0$ and $y_4 = 0$, we obtain

$$\begin{cases} \hat{\pi}_A = \pi_{A1} = 2.0 \\ \hat{\pi}_A = \pi_{A2} = 1.5y_1 + 3.0y_2 + 3.0y_3 \\ \hat{\pi}_A = \pi_{A3} = 2.0y_2 + 4.0y_3 \\ \hat{\pi}_A = \pi_{A4} = 2.5y_3 \\ \hat{\pi}_A > \pi_{A5} = 0.0 \\ \hat{\pi}_B = \pi_{B1} = 0.5x_2 + 1.0x_3 + 1.0x_4 \\ \hat{\pi}_B = \pi_{B2} = 1.0x_3 + 2.0x_4 \\ \hat{\pi}_B = \pi_{B3} = 1.5x_4 \\ \hat{\pi}_B > \pi_{B4} = 0.0 \end{cases}$$

This time, we can show $x_4 = 0$ by contradiction in the same way. Thus, we have $\pi_{B3} = 0.0$.

When $\hat{\pi}_B = \pi_{B3} = 0.0$, we will obtain the same mixed-strategy Nash equilibrium as i). When $\hat{\pi}_B > \pi_{B3} = 0.0$, we have $y_3 = 0$ from the Fundamental Theorem.

By setting $x_4 = 0$ and $y_3 = 0$ this time, we obtain

$$\begin{cases} \hat{\pi}_A = \pi_{A1} = 2.0 \\ \hat{\pi}_A = \pi_{A2} = 1.5y_1 + 3.0y_2 \\ \hat{\pi}_A = \pi_{A3} = 2.0y_2 \\ \hat{\pi}_A > \pi_{A4} = 0.0 \\ \hat{\pi}_A > \pi_{A5} = 0.0 \\ \hat{\pi}_B = \pi_{B1} = 0.5x_2 + 1.0x_3 \\ \hat{\pi}_B = \pi_{B2} = 1.0x_3 \\ \hat{\pi}_B > \pi_{B3} = 0.0 \\ \hat{\pi}_B > \pi_{B4} = 0.0 \end{cases}$$

Again, we can show $x_3 = 0$ by contradiction and have $\pi_{B2} = 0.0$. When

$\hat{\pi}_B = \pi_{B2} = 0.0$, we will obtain the same mixed-strategy Nash equilibrium as i). When $\hat{\pi}_B > \pi_{B2} = 0.0$, we have $y_2 = 0$ from the Fundamental Theorem, which determines the mixed-strategy Nash equilibrium for PPS2 as $y = [1, 0, 0, 0]$.

By setting $x_3 = 0$, $y_1 = 1$ and $y_2 = 0$, we obtain

$$\left\{\begin{array}{ccccc} & & \pi_{A1} & = & 2.0 \\ & & \pi_{A2} & = & 1.5 \\ \hat{\pi}_A & > & \pi_{A3} & = & 0.0 \\ \hat{\pi}_A & > & \pi_{A4} & = & 0.0 \\ \hat{\pi}_A & > & \pi_{A5} & = & 0.0 \\ \hat{\pi}_B & = & \pi_{B1} & = & 0.5x_2 \\ \hat{\pi}_B & > & \pi_{B2} & = & 0.0 \\ \hat{\pi}_B & > & \pi_{B3} & = & 0.0 \\ \hat{\pi}_B & > & \pi_{B4} & = & 0.0 \end{array}\right.$$

Since $\pi_{A1} > \pi_{A2}$, we conclude that $\hat{\pi}_A = \pi_{A1} > \pi_{A2}$ and $x = [1, 0, 0, 0, 0]$ is a mixed-strategy Nash equilibrium for PPS1. Under this equilibrium, $\hat{\pi}_B = \pi_{Bi} = 0.0$ for $^\forall i$. However, it contradicts the condition $\hat{\pi}_B > \pi_{B4} = 0.0$ that we set for ii). Therefore, there isn't a mixed-strategy Nash equilibrium under which PPS2 obtains a positive payoff.

In conclusion, in the non-repeated auction of our assumed model, (x^*, y^*) is a mixed-strategy Nash equilibrium if and only if

(1) $x^* = [1, 0, 0, 0, 0]$,and

(2) $\hat{\pi}_A = \pi_{A1} \geq \pi_{Ai} \Leftrightarrow [Ay^*]_1 \geq [Ay^*]_i$,for $i = 2, 3, 4, 5$

Chapter 3

Markov Chain Analysis

3.1 Formulation

In this chapter, the auction assumed in the previous chapter is repeated infinitely, and the dynamics of the game are investigated. A Markov chain is used to model the repeated game. This is a common way to model repeated games [4], but players can carry over information only from the last game. Some dynamic characteristics are lost by this assumption, but we are still able to derive some insight.

Now, assume that players in the electricity market change their bid price in the next auction according to the following rule:

If player wins,

$$\begin{cases} \text{Raise its bid price} & : \text{ probability} = a \\ \text{Stay} & : \text{ probability} = 1 - a \end{cases}$$

If player loses,

$$\begin{cases} \text{Lower its bid price} & : \text{ probability} = b \\ \text{Stay} & : \text{ probability} = 1 - b \end{cases}$$

If players tie, each independently will

$$\begin{cases} \text{Lower its bid price} & : \ \text{probability} = c \\ \text{Stay} & : \ \text{probability} = 1 - c \end{cases}$$

When players raise or lower their bid price, only a one-step change, for example from p_1 = 7.0 to p_1 = 8.0, is allowed in the model.

In this assumption, when a player wins, it stays at the same strategy or chooses a more profitable strategy. However, this more profitable strategy is expected to have a lower probability of winning at the next auction. When a player loses, it stays at the same strategy or chooses a less profitable strategy which is expected to have a higher probability of winning at the next auction. When players tie, they behave as they do when they lose, but they are more likely to stay at the same strategy, that is $c < b$.

3.2 Analysis of the Markov Process

Figure 3-1 shows the Markov chain of this repeated auction. In the figure, the state consists of PPS1's and PPS2's strategies, and is shown as (p_1, p_2). The states, (7.0, 8.0) and (8.0, 8.0), constitute the only absorption class in the Markov chain, and the other states are transient states. Thus, after enough repeated auctions, PPS1 is bidding at p_1 = 7.0 or 8.0, and PPS2 is bidding at p_2 = 8.0. In this steady state, the probability that PPS1 bids at p_1 = 7.0 is $\frac{c}{a+c}$, and the probability that it bids at p_1 = 8.0 is $\frac{a}{a+c}$. The Markov chain analysis suggests that the equilibrium is at p_1 = 7.0 even in the repeated auction.

When we look at Figure 3-1 carefully, we notice that the states are separated into the four zones shown in Figure 3-2. In Zone1, PPS2 loses to or ties with PPS1, and its bid price is 8.0. In Zone2, when PPS1 loses to or ties with PPS2, its bid price is 9.0. When PPS2 loses to or ties with PPS1, its bid price is 9.0. Similarly, in Zone3, when PPS1 or PPS2 loses or ties, its bid price is 10.0. In Zone4, it is 11.0.

Zone1 is accessible from all other zones, and Zone2 is accessible from Zone3 and

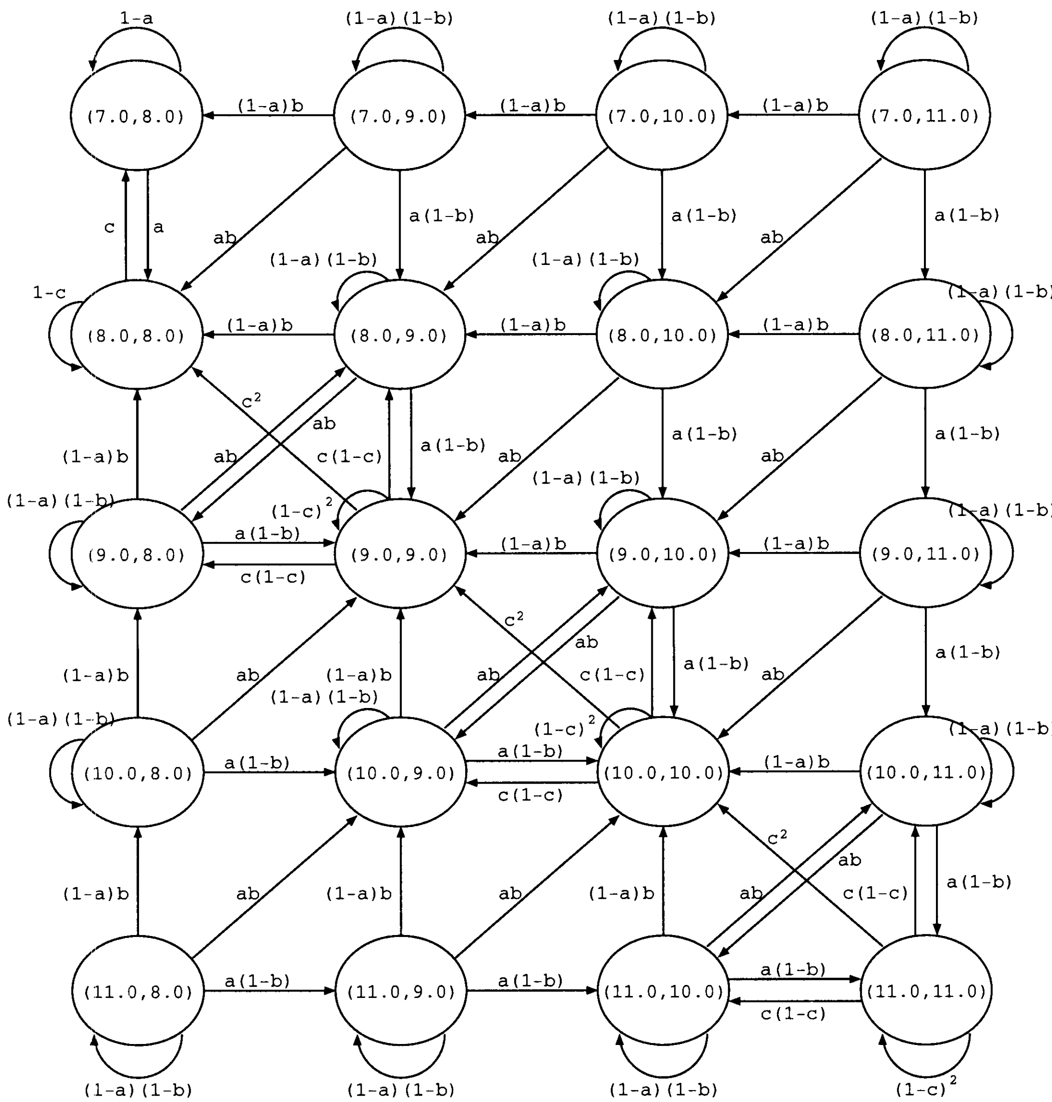

Figure 3-1: Markov chain of repeated auction.

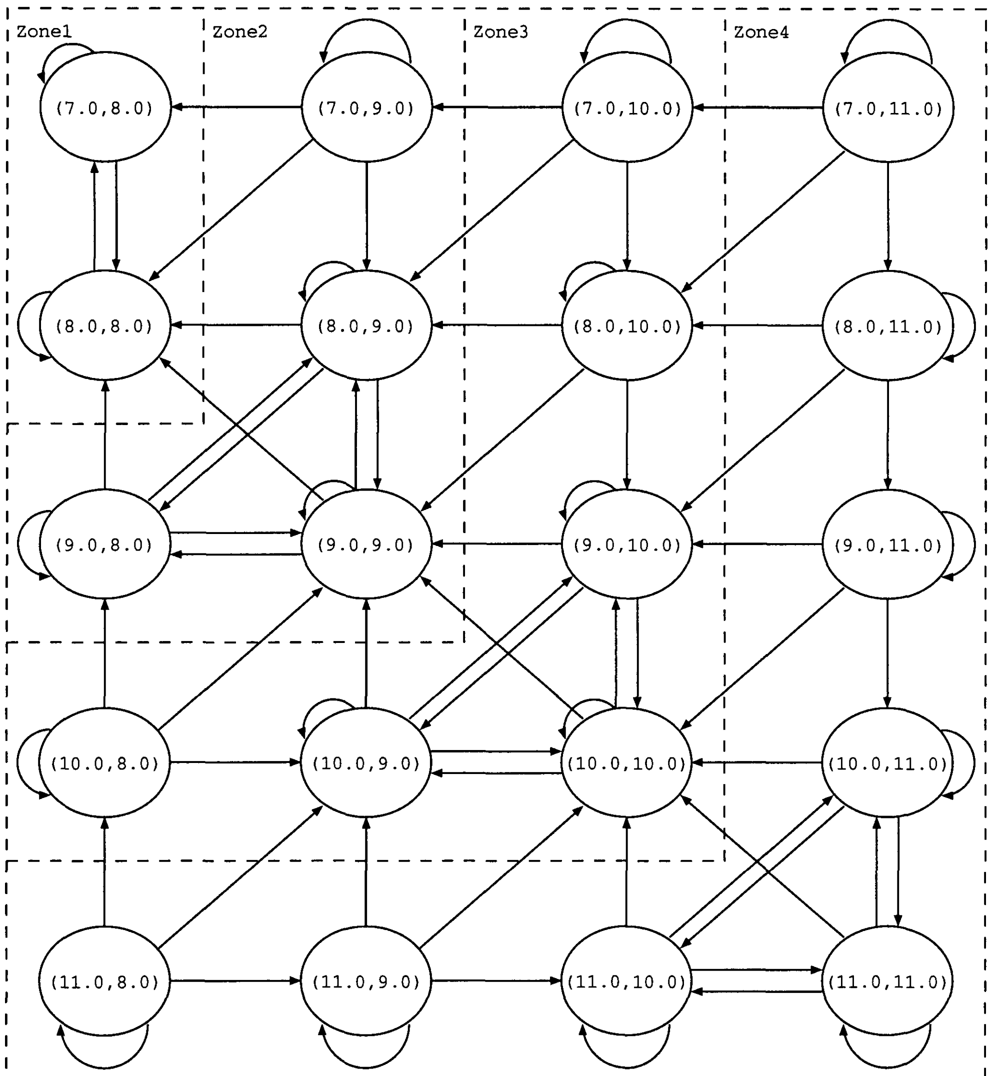

Figure 3-2: Zones of the Markov chain.

Zone4. Similarly, Zone3 is accessible only from Zone4, and Zone4 is not accessible from any other zones. Thus, if the current state is in Zone4, it will be eventually absorbed in Zone1 via Zone3 and Zone2. This transition from zone to zone happens when a loser lowers its bid price. It also happens when both players lower their bid price after they tie.

Therefore, when the starting state is in Zone4, the probability b plays an important role in determining whether the players can reach Pareto efficient states (p_1, p_2) = (10.0, 11.0), (11.0, 10.0), (11.0, 11.0) or not. For example, starting from (p_1, p_2) = (11.0, 8.0) in Zone4, the probability that PPS1 and PPS2 go to Zone3 at the next auction is

$$(1-a)b + ab = b$$

Thus, when b is closer to 0, they are more likely to stay in Zone4. It means PPS1 can patiently wait and see how PPS2 behaves when it wins. Suppose that they go to (p_1, p_2) = (11.0, 9.0) in Zone4 after some auctions. The probability that they go to Zone3 at the next auction is also b. The same is true when they start from (p_1, p_2) = (7.0, 11.0), in which PPS1 wins over PPS2. From the example, we can see that when b is closer to 0, they are more likely to be able to reach Pareto efficient states. This suggests that, in order to achieve Pareto efficiency, the patience to wait until the other player raises its bid price is important.

This situation matches the published experience with repeated Prisoner's Dilemma experiments. In such an experiment, a prisoner, say Prisoner 1, often cooperates first, and looks at how the other prisoner, Prisoner 2, behaves. If Prisoner 1 is patient enough to wait until Prisoner 2 cooperates, they can reach the Pareto efficient strategy together. Patience is the key to achieving Pareto efficiency in both situations.

Also, the process of getting from the Pareto efficient states to the Nash equilibrium states is similar in both situations. The Pareto efficient states are unstable, and eventually break down to the Nash equilibrium. Just as each prisoner betrays the other prisoner at the Nash equilibrium, both players in our Markov process eventually end up bidding the low price in Zone1.

In repeated Prisoner's Dilemma experiments, prisoners often tacitly renegotiate to accomplish Pareto efficiency from the non-Pareto optimal Nash equilibrium, but this renegotiation is not modeled in this Markov process. This is the reason why the states, $(p_1,\ p_2) = (7.0,\ 8.0),\ (8.0,\ 8.0)$, constitute the only absorption class in this Markov process.

3.3 Taking a Firm Stand

What if the players never lower their bid price when they tie, that is $c = 0$? Figure 3-3 shows the Markov process when $c = 0$. Now, all four states in which the players tie, $(p_1,\ p_2) = (8.0,\ 8.0),\ (9.0,\ 9.0),\ (10.0,\ 10.0),\ (11.0,\ 11.0)$, are absorption states.

Given a starting state, the probability that PPS1 and PPS2 end up in each absorption state can be calculated from Figure 3-3. For example, the calculation of the probability that starting from $(p_1,\ p_2) = (11.0,\ 8.0)$, PPS1 and PPS2 end up in the absorption state $(p_1,\ p_2) = (11.0,\ 11.0)$, is shown here. This probability is same as the probability that, for c $\neq 0$ and starting from $(p_1,\ p_2) = (11.0,\ 8.0)$, PPS1 and PPS2 go through the Pareto efficient state $(p_1,\ p_2) = (11.0,\ 11.0)$, because the state can only go from Zone4 to Zone1.

Let $P(p_1, p_2)$ denote the probability that starting from $(p_1,\ p_2)$, PPS1 and PPS2 end up in the absorption state $(p_1,\ p_2) = (11.0,\ 11.0)$. Then, the solution is given as $P(11.0, 8.0)$. Using transition probabilities, $P(p_1, p_2)$ satisfies following equations:

$$\begin{cases} P(10.0,11.0) &= a(1-b) & + & (1-a)(1-b)P(10.0,11.0) & + & abP(11.0,10.0) \\ P(11.0,8.0) &= a(1-b)P(11.0,9.0) & + & (1-a)(1-b)P(11.0,8.0) & & \\ P(11.0,9.0) &= a(1-b)P(11.0,10.0) & + & (1-a)(1-b)P(11.0,9.0) & & \\ P(11.0,10.0) &= a(1-b) & + & (1-a)(1-b)P(11.0,10.0) & + & abP(10.0,11.0) \end{cases}$$

Solving this system of equations, the solution is given as

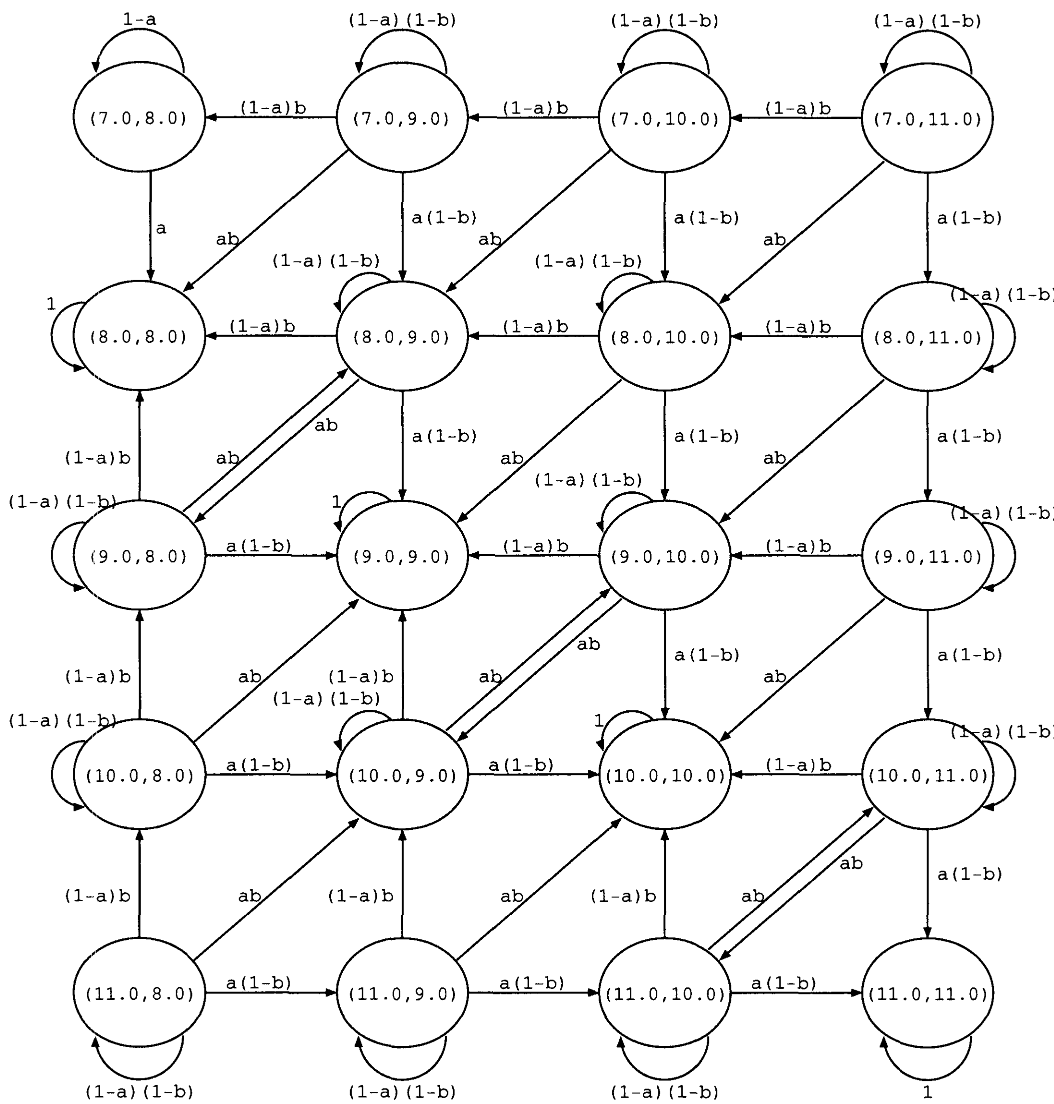

Figure 3-3: Markov Process with c = 0

$$P(11.0, 8.0) = \frac{a^3(1-b)^3}{(a+b-ab)^2(a+b-2ab)}$$

This probability, $P(11.0, 8.0)$, is shown in Figure 3-4. We can see that the probability is large when a is close to 1 and b is close to 0, and the decline of the probability is sharp when a and b go away from their optimal ranges. The smaller a is, the sharper the decline of the probability is when b increases from 0. Because of the sharpness of this decline, if the given a and b are in the area where the surface of $P(11.0, 8.0)$ is flat, which accounts for more than half of ab-plane, we see that PPS1 and PPS2 are not likely to be absorbed in the state $(p_1,\ p_2)$ = (11.0, 11.0).

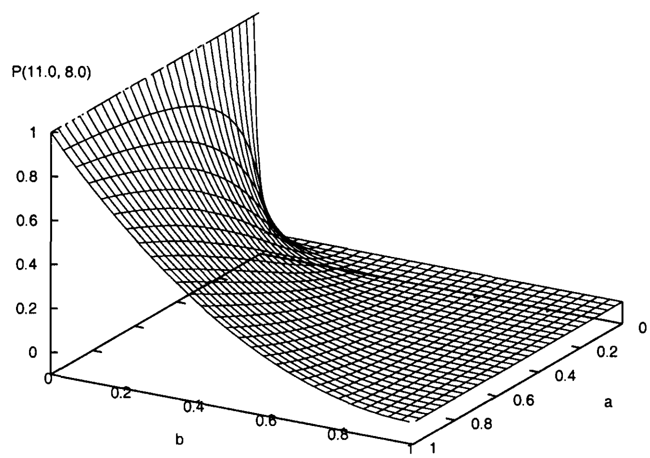

Figure 3-4: Probability to be absorbed to (11.0, 11.0).

Starting from $(p_1,\ p_2)$ = (11.0, 8.0), the probabilities that PPS1 and PPS2 end up in other absorption states can also be calculated. Let $Pr(p_1)$ denote these probabilities that PPS1 and PPS2 end up with the strategy p_1 (which is equal to p_2). Then, $Pr(11.0) = P(11.0, 8.0)$.

$$
\begin{cases}
Pr(8.0) &= \dfrac{(1-a)^3b^3}{(a+b-ab)^2(a+b-2ab)} \\[2ex]
Pr(9.0) &= \dfrac{ab^2(1-a)(3-a-b)}{(a+b-ab)^2(a+b-2ab)} \\[2ex]
Pr(10.0) &= \dfrac{a^2b(1-b)(3-a-b)}{(a+b-ab)^2(a+b-2ab)} \\[2ex]
Pr(11.0) &= \dfrac{a^3(1-b)^3}{(a+b-ab)^2(a+b-2ab)}
\end{cases}
$$

According to the magnitude of $Pr(p_1)$, we can see to which states PPS1 and PPS2 are most likely to be absorbed, starting from $(p_1, p_2) = (11.0, 8.0)$. The absorption state with the largest $Pr(p_1)$ among $Pr(8.0)$, $Pr(9.0)$, $Pr(10.0)$, and $Pr(11.0)$ is shown in Figure 3-5. Since $Pr(8.0)$, $Pr(9.0)$, and $Pr(10.0)$ have the same sharpness of decline as $Pr(11.0)$, when the given a and b are away from their optimal ranges, the probability that PPS1 and PPS2 end up in the absorption state shown in Figure 3-5 is relatively high.

The boundary line between (8.0, 8.0) and (9.0, 9.0) is

$$b = \frac{2a+1}{a^2-a+1} - 1$$

The boundary line between (9.0, 9.0) and (10.0, 10.0) is

$$b = a$$

The boundary line between (10.0, 10.0) and (11.0, 11.0) is

$$a = \frac{2b+1}{b^2-b+1} - 1$$

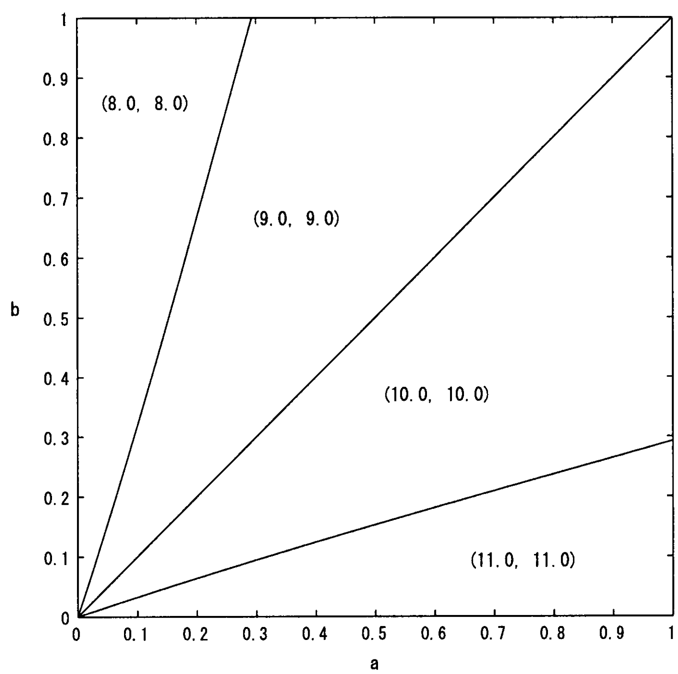

Figure 3-5: Most likely absorption states.

Chapter 4

Replicator Dynamics

4.1 Discrete Time Replicator Equation

The replicator equation [4][8] is also commonly used to analyze the dynamics of repeated games, especially in biology. The replicator equation for the probability that the player uses the i-th strategy is written as

$$\dot{x}_i = \alpha x_i \left(\pi_i - \hat{\pi}\right)$$

where

x_i	:	the probability that the player uses the i-th strategy
$\dot{x}_i$	:	time derivative of x_i
α	:	coefficient for the replicator equation
π_i	:	payoff for the player using the i-th strategy
$\hat{\pi}$	:	average payoff over all strategies
		($= \pi_1 x_1 + \pi_2 x_2 + \cdots + \pi_n x_n$, for n strategies)

Unlike in previous chapters, players can use a mixed strategy in this chapter, and during the repeated auction, they change the probability that they use each strategy.

Now, considering one auction as one time-step, a discrete-time replicator equation can be derived. The strategies are sorted in ascending order according to their payoff, so when there are n strategies, then $\pi_1 \le \pi_2 \le \cdots \le \pi_n$. During the repeated auction, the player using the i-th strategy may notice that the j-th strategy is better than the i-th strategy, and switches to the j-th strategy. The probability that the player switches to the j-th strategy in the next auction is given as follows:

$$p_{ij} = \begin{cases} \beta x_j \left(\pi_j - \pi_i\right) & , if\ \pi_j > \pi_i \\ 0 & , otherwise \end{cases}$$

Here, β, the coefficient for the replicator equation, has to be small enough to guarantee that every p_{ij} is smaller than or equal to 1, and $\sum_{j=i+1}^{n} p_{ij} \leq 1$ for $\forall i$. In order to satisfy the condition, $\beta = 0.02$ is used in this chapter.

The more the player uses the j-th strategy and the larger the difference between π_i and π_j is, the more likely the player is to switch to the j-th strategy.

Given the probability $x_i(t)$ that the player uses the i-th strategy at time t, we can write:

$$\begin{aligned} x_i(t+1) &= x_i(t) + \sum_{j=1}^{i-1} p_{ji} x_j(t) - x_i(t) \sum_{j=i+1}^{n} p_{ij} \\ &= x_i(t) + x_i(t) \sum_{j=1}^{i} \beta \left(\pi_i - \pi_j\right) x_j(t) - x_i(t) \sum_{j=i+1}^{n} \beta \left(\pi_j - \pi_i\right) x_j(t) \\ &= x_i(t) + \beta x_i(t) \sum_{j=1}^{n} x_j(t) \left(\pi_i - \pi_j\right) \\ &= x_i(t) + \beta x_i(t) \left(\pi_i - \hat{\pi}\right) \end{aligned}$$

Applying this discrete-time replicator equation, the model in Chapter 2 can be analyzed. The limitations of the Markov chain analysis in representing changing strategies no longer apply. The auction is repeated until the dynamics become apparent. Since there are two players to be updated by their respective replicator equations, the update takes place in turns. MATLAB® is used for calculation, and the script is shown in Appendix A.1.

Starting from the uniform distribution, after 100 auctions the probabilities that PPS1 and PPS2 use the various strategies are given in the table below:

Strategy	7.0	8.0	9.0	10.0	11.0
PPS1	0.1568	0.5041	0.2758	0.0554	0.0079
PPS2	-	0.3316	0.3473	0.2022	0.1189

The row labeled **Strategy** represents the bid prices of PPS1 and PPS2.

After 2000 auctions, the probabilities that PPS1 and PPS2 use the various strategies looks quite different:

Strategy	7.0	8.0	9.0	10.0	11.0
PPS1	1.0000	0.0000	0.0000	0.0000	0.0000
PPS2	-	0.9827	0.0112	0.0039	0.0022

Thus, PPS1 eventually finds that the pure strategy $p_1 = 7.0$ is the best strategy, which is the Nash equilibrium in the non-repeated auction, and PPS1 and PPS2 don't find Pareto efficient strategies. Thus, the replicator dynamics also suggests that the equilibrium exists at $p_1 = 7.0$ in the repeated auction.

4.2 Evolution of Strategy to the Nash Equilibrium

Now, what if the first auction starts from a probability distribution that is close to Pareto efficient? For instance, suppose:

Strategy	7.0	8.0	9.0	10.0	11.0
PPS1	0.0001	0.0001	0.0001	0.0001	0.9996
PPS2	-	0.0001	0.0001	0.0001	0.9997

After 10000 auctions, the probability distribution converges to

Strategy	7.0	8.0	9.0	10.0	11.0
PPS1	1.0000	0.0000	0.0000	0.0000	0.0000
PPS2	-	1.0000	0.0000	0.0000	0.0000

Here is how this evolution of strategy progresses. At the first auction, the payoff for PPS1 is

$$\begin{bmatrix} \pi(7.0) \\ \pi(8.0) \\ \pi(9.0) \\ \pi(10.0) \\ \pi(11.0) \end{bmatrix} = \begin{bmatrix} 2.0000 \\ 2.9999 \\ 3.9994 \\ 4.9988 \\ 2.9991 \end{bmatrix},$$

where $\pi(7.0)$ represents the payoff when PPS1 uses the strategy $p_1 = 7.0$. As can be seen, $\pi(10.0)$ is larger than $\pi(11.0)$. Since PPS2 adopts the strategy $p_2 = 11.0$, with a probability of 0.9997, PPS1 ties with PPS2 with a probability of 0.9997 when PPS1 adopts the strategy $p_1 = 11.0$. This brings the payoff $u_1 = 3.0$ to PPS1. When PPS1 adopts the strategy $p_1 = 10.0$, PPS1 wins over PPS2 with a probability of 0.9997, and earns the payoff $u_1 = 5.0$. Pareto efficiency doesn't guarantee optimality to the individuals, and in this case PPS1 can obtain a better payoff by lowering its bid price. For this reason, PPS1 shifts its strategy from $p_1 = 11.0$ to $p_1 = 10.0$. If PPS1 knew at this point that it could obtain only $u_1 = 2.0$ at the Nash equilibrium, PPS1 might settle for $u_1 = 3.0$ by choosing $p_1 = 11.0$. However, PPS1 doesn't know where the Nash equilibrium is at first, so there is not any deterrence for PPS1 to shift its strategy from $p_1 = 11.0$ to $p_1 = 10.0$.

As for PPS2, the payoff at the first auction is

$$\begin{bmatrix} \pi(8.0) \\ \pi(9.0) \\ \pi(10.0) \\ \pi(11.0) \end{bmatrix} = \begin{bmatrix} 0.9999 \\ 1.9995 \\ 2.9989 \\ 1.9992 \end{bmatrix}$$

The same explanation can be made for PPS2. PPS2 obtains the payoff $u_2 = 0.0$ at the Nash equilibrium, so shifting its strategy from $p_2 = 11.0$ to $p_2 = 10.0$ is actually a risky evolution of strategy for PPS2, because it can lead PPS1 to further lower its bid price. PPS2, however, doesn't know that its cost is higher than that of PPS1, and PPS2 shifts its strategy from $p_2 = 11.0$ to $p_2 = 10.0$.

This observation suggests the difficulty of keeping players at Pareto efficient strategies when there are many sellers in the auction. In the model assumed in this thesis, players don't have information about other players. In a real auction, however, players might have information about other players from past records or the conditions it shares with other players. This information can give a clue for the players to predict the Nash equilibrium. Just as PPS1 might settle for $u_1 = 3.0$ by choosing $p_1 = 11.0$ if it knew the Nash equilibrium, the information could help to prevent the breakdown to the Nash equilibrium. The more players that take part in the auction, the more difficult it becomes to estimate the Nash equilibrium, which causes players to drift to the Nash equilibrium.

After 500 auctions, the evolution of strategy from $(p_1,\ p_2) = (11.0,\ 11.0)$ to $(p_1,\ p_2) = (10.0,\ 10.0)$ can be clearly seen in the next table:

Strategy	7.0	8.0	9.0	10.0	11.0
PPS1	0.0000	0.0000	0.0161	0.9839	0.0000
PPS2	-	0.0000	0.1198	0.8506	0.0296

At this point, the payoff for PPS1 is

$$\begin{bmatrix} \pi(7.0) \\ \pi(8.0) \\ \pi(9.0) \\ \pi(10.0) \\ \pi(11.0) \end{bmatrix} = \begin{bmatrix} 2.0000 \\ 3.0000 \\ 3.7627 \\ 2.2796 \\ 0.0914 \end{bmatrix}$$

The payoff for PPS2 is

$$\begin{bmatrix} \pi(8.0) \\ \pi(9.0) \\ \pi(10.0) \\ \pi(11.0) \end{bmatrix} = \begin{bmatrix} 1.0000 \\ 1.9839 \\ 1.4758 \\ 0.0000 \end{bmatrix}$$

Since $\pi(9.0)$ is larger than $\pi(10.0)$ for both PPS1 and PPS2, they shift their strategies from $(p_1, p_2) = (10.0, 10.0)$ to $(p_1, p_2) = (9.0, 9.0)$ this time. The repetition of this evolution of strategy eventually leads PPS1 and PPS2 to the Nash equilibrium.

4.3 Competition Among Three Players

By using replicator equations, a repeated auction with more than two sellers can be analyzed. In this section, we observe how the evolution of strategy changes under the intensified competition.

Example

A company, PPS0, announces to buy d MWh of electricity with its maximum acceptable price $P = 12.0$ at the auction. This time, three companies, PPS1, PPS2 and PPS3, bid to sell their electricity to PPS0. Their bid prices to PPS0 are p_1, p_2, and p_3, respectively, and their costs to supply d to PPS0 are $c_1 = 5.0$, $c_2 = 6.0$, and $c_3 = 7.0$, respectively. Here, $c_1 < p_1 < P$, $c_2 < p_2 < P$, and $c_3 < p_3 < P$.

The acceptable bid price for PPS1 is $c_1 + 1.0 \times i$ (i = 1,2,3,4,5,6), and is $c_2 + 1.0 \times i$ (i = 1,2,3,4,5) for PPS2. For PPS3, it is $c_3 + 1.0 \times i$ (i = 1,2,3,4).

When two players win with the same bid price, for example PPS1 and PPS2 win with $p_1 = p_2$, PPS1 and PPS2 both sell $\frac{d}{2}$ MWh of electricity to PPS0 at the price $\frac{p_1}{2}$. The cost to sell $\frac{d}{2}$ MWh of electricity is $\frac{c_1}{2}$ and $\frac{c_2}{2}$ respectively for PPS1 and PPS2.

If all three players tie at the same bid price, the three players sell $\frac{d}{3}$ MWh of electricity to PPS0 at the price $\frac{p_1}{3}$. The cost to sell $\frac{d}{3}$ MWh of electricity is $\frac{c_1}{3}$, $\frac{c_2}{3}$, and $\frac{c_3}{3}$ for PPS1, PPS2, and PPS3, respectively.

The payoff for PPS1 is

$$u_1 = \begin{cases} p_1 - c_1 & , if\ p_1 < p_2\ and\ p_1 < p_3 \\ \\ \dfrac{p_1 - c_1}{2} & , if\ p_1 = p_2 < p_3\ or\ p_1 = p_3 < p_2 \\ \\ \dfrac{p_1 - c_1}{3} & , if\ p_1 = p_2 = p_3 \\ \\ 0 & , otherwise \end{cases}$$

The payoff for PPS2 and PPS3 is given in the same fashion as for PPS1.

If this auction is a non-repeated game, one of the Nash equilibria is $p_1 = 6.0$, since PPS1 can always undercut the bids of PPS2 and PPS3 by bidding at $p_1 = 6.0$. In that Nash equilibrium, $(u_1, u_2, u_3) = (1.0, 0.0, 0.0)$. $p_1 = p_2 = 7.0$ is the other Nash equilibrium of this non-repeated game. As PPS1 receives the same payoff in this equilibrium, PPS1 doesn't have an incentive to lower its bid price once the dynamics settles at the equilibrium. In this case, $(u_1, u_2, u_3) = (1.0, 1.0, 0.0)$.

Now, we go back to the repeated game. Since three players bid to sell their electricity, the payoff matrix becomes three-dimensional. The plane $p_3 = 8.0$ of the payoff matrix is shown in Table 4.1.

In Table 4.1, the payoff is shown as (u_1, u_2, u_3). In the 3×3 subspace on the bottom-right corner of Table 4.1, $(u_1, u_2, u_3) = (0.0, 0.0, 1.0)$ because $p_1 > 8.0$, $p_2 > 8.0$, and $p_3 = 8.0$. The other three planes, $p_3 = 9.0$, 10.0, and 11.0, of the payoff matrix also need to be calculated for the analysis. The MATLAB script for the replicator analysis with three sellers is shown in Appendix A.2, and it includes the other three planes of the payoff matrix.

PPS1's Strategy (p_1) \ PPS2's Strategy (p_2)	7.0	8.0	9.0	10.0	11.0
6.0	(1.0, 0.0, 0.0)	(1.0, 0.0, 0.0)	(1.0, 0.0, 0.0)	(1.0, 0.0, 0.0)	(1.0, 0.0, 0.0)
7.0	(1.0, 0.5, 0.0)	(2.0, 0.0, 0.0)	(2.0, 0.0, 0.0)	(2.0, 0.0, 0.0)	(2.0, 0.0, 0.0)
8.0	(0.0, 1.0, 0.0)	(1.0, 2/3, 1/3)	(1.5, 0.0, 0.5)	(1.5, 0.0, 0.5)	(1.5, 0.0, 0.5)
9.0	(0.0, 1.0, 0.0)	(0.0, 1.0, 0.5)	(0.0, 0.0, 1.0)	(0.0, 0.0, 1.0)	(0.0, 0.0, 1.0)
10.0	(0.0, 1.0, 0.0)	(0.0, 1.0, 0.5)	(0.0, 0.0, 1.0)	(0.0, 0.0, 1.0)	(0.0, 0.0, 1.0)
11.0	(0.0, 1.0, 0.0)	(0.0, 1.0, 0.5)	(0.0, 0.0, 1.0)	(0.0, 0.0, 1.0)	(0.0, 0.0, 1.0)

Table 4.1: Payoff matrix ($p_3 = 8.0$).

Starting from the uniform distribution, after 100 auctions, the probability that three sellers use each strategy is:

Strategy	6.0	7.0	8.0	9.0	10.0	11.0
PPS1	0.0996	0.4467	0.3307	0.0814	0.0270	0.0147
PPS2	-	0.3151	0.3349	0.1679	0.1029	0.0792
PPS3	-	-	0.3214	0.2689	0.2181	0.1916

After 1500 auctions, the probability that the three sellers use each strategy is:

Strategy	6.0	7.0	8.0	9.0	10.0	11.0
PPS1	0.0169	0.9831	0.0000	0.0000	0.0000	0.0000
PPS2	-	1.0000	0.0000	0.0000	0.0000	0.0000
PPS3	-	-	0.3688	0.2544	0.2011	0.1756

Therefore, their bidding strategy eventually converges to the Nash equilibrium of the non-repeated game. Since PPS2 chooses the strategy $p_2 = 7.0$ with probability 1.0000, the strategies $p_1 = 6.0$ and $p_1 = 7.0$ bring the same payoff, $u_1 = 1.0$, for PPS1. Thus, any state in which PPS1 chooses $p_1 = 6.0$ and $p_1 = 7.0$ with a probability of x and $1 - x$, respectively, can be an equilibrium for $\forall x \in [0.0,\ 1.0]$. In this result, PPS1 chooses $p_1 = 6.0$ with a probability of 0.0169 and $p_1 = 7.0$ with a probability of 0.9831.

Compared with the result of the two-seller case, the bidding strategy in the three-seller case converges faster to the Nash equilibrium. After 100 auctions, in the two-seller case, PPS1 chooses the strategy $p_1 = 8.0$ with the highest probability, 0.5041.

In the three-seller case, PPS1 chooses the strategy $p_1 = 7.0$ with the highest probability, 0.4467. This result suggests PPS1 shifts its strategy faster due to the intense competition.

Chapter 5

First Round of the Auction

In previous chapters, we have analyzed the second round of the electricity auction in Japan. In this chapter, we analyze the first round of the electricity auction, where buyers bid to seller announcements. We will show that there exists a Nash equilibrium which is symmetrical to the one we have derived for the second round of the auction.

5.1 Non-Repeated Game Analysis

In this section, the first round of the auction is restricted to be held only once.

Assumed Model

PPS0 announces its intent to sell d MWh of electricity with its minimum acceptable price P at the auction, and PPS1 and PPS2 bid to buy d MWh of electricity from PPS0. Their respective bid prices to PPS0 are p_1 and p_2, which must be higher than P. If PPS1 and PPS2 produce electricity by themselves, their costs to generate d are c_1 and c_2, respectively. When they cannot meet their demands by themselves, their costs are equal to the market reserve prices, P_{m1} and P_{m2}, by which price they can procure the needed electricity from large utilities. Here, $P < p_1 < c_1 \leq P_{m1}$, $P < p_2 < c_2 \leq P_{m2}$, and assume without loss of generality that $c_1 > c_2$.

For this model, there is a unique pure-strategy Nash equilibrium, in which PPS1 bids with the price $p_1 = c_2$ and wins the auction. The proof is given as follows. If PPS1 bids with $p_1 < c_2$, PPS2 can win the auction by $p_2 > p_1$. However, if PPS1 bids with $p_1 \geq c_2$, PPS1 always wins the auction because $p_2 < c_2$. Among the choices for PPS1 within $c_1 > p_1 \geq c_2$, $p_1 = c_2$ brings the best profit to PPS1. Thus, bidding with $p_1 = c_2$ is the best strategy for PPS1 regardless of PPS2's bid.

We can also derive this Nash equilibrium by the iterated elimination of dominated strategies. Let $c_1 = 12.0$, $c_2 = 10.0$, and $P = 5.0$. Then the acceptable bid prices for PPS1 are 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, and PPS2's bid price is one of 6.0, 7.0, 8.0, 9.0. PPS1 can always win when $p_1 = 10.0$ or 11.0, but since $p_1 = 11.0$ is obviously strictly dominated by $p_1 = 10.0$, assume that PPS1's bid price is one of 6.0, 7.0, 8.0, 9.0, 10.0. The other conditions are same as those in the example in Section 2.3.

Table 5.1 is the payoff matrix for this model. Note that this payoff matrix can be obtained by rotating the second-round payoff matrix in Table 2.1 by 180 degrees. Since the upper left strategy survives the iterated elimination from Table 2.1, we expect the bottom right strategy survives the iterated elimination from Table 5.1. Starting from the payoff matrix shown in Table 5.1, we reach the Nash equilibrium (p_1, p_2) = (10.0, 9.0), as we expected, by following elimination trees shown in Figure 5-1 and Figure 5-2, regardless of the order of eliminations. At this Nash equilibrium, PPS1 wins the auction with the bid price $p_1 = c_2$, and this result matches the equilibrium derived by the theoretical reasoning.

		PPS2's Strategy (p_2)			
		6.0	7.0	8.0	9.0
PPS1's Strategy (p_1)	6.0	(3.0, 2.0)	(0.0, 3.0)	(0.0, 2.0)	(0.0, 1.0)
	7.0	(5.0, 0.0)	(2.5, 1.5)	(0.0, 2.0)	(0.0, 1.0)
	8.0	(4.0, 0.0)	(4.0, 0.0)	(2.0, 1.0)	(0.0, 1.0)
	9.0	(3.0, 0.0)	(3.0, 0.0)	(3.0, 0.0)	(1.5, 0.5)
	10.0	(2.0, 0.0)	(2.0, 0.0)	(2.0, 0.0)	(2.0, 0.0)

Table 5.1: Payoff matrix for the first round of the auction.

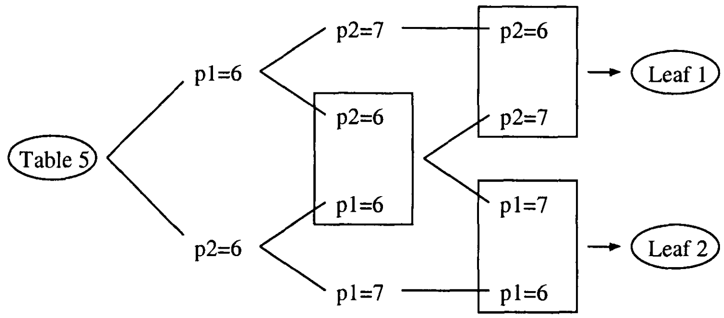

Figure 5-1: Elimination tree for the first round of the auction.

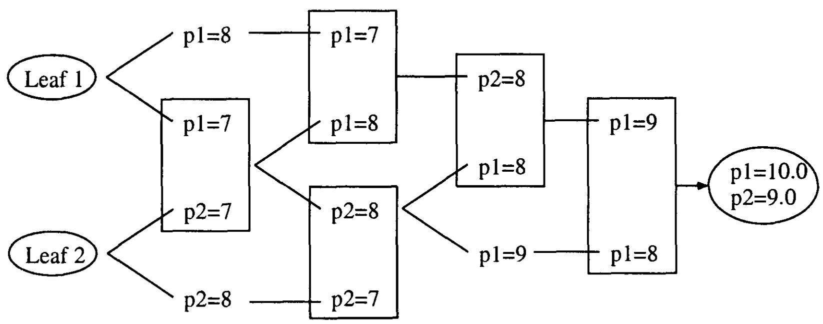

Figure 5-2: Elimination tree after Figure 5-1.

5.2 Markov Chain Analysis

For the Markov chain analysis, assume that players change their strategies according to the following rule:

If player wins,

$$\begin{cases} \text{Lower its bid price} & : \text{ probability} = a \\ \text{Stay} & : \text{ probability} = 1 - a \end{cases}$$

If player loses,

$$\begin{cases} \text{Raise its bid price} & : \text{ probability} = b \\ \text{Stay} & : \text{ probability} = 1 - b \end{cases}$$

If players tie, each independently will

$$\begin{cases} \text{Raise its bid price} & : \ \text{probability} = c \\ \text{Stay} & : \ \text{probability} = 1 - c \end{cases}$$

In this assumption, when a player wins, it stays at the same strategy or chooses a more profitable strategy. However, this more profitable strategy is expected to have a lower probability of winning at the next auction. When a player loses, it stays at the same strategy or chooses a less profitable strategy which is expected to have a higher probability of winning at the next auction. Thus, this assumption corresponds to that for the analysis of the second round of auction stated in Chapter 3.

Figure 5-3 shows the Markov chain of this repeated auction. Again, we can see that this Markov chain can be obtained by rotating the Markov chain in Figure 3-1 by 180 degrees. In Figure 5-3, the states, $(p_1,\ p_2) = (9.0,\ 9.0)$ and $(10.0,\ 9.0)$, constitute the only absorption class in the Markov chain, and the other states are transient states. Thus, after enough repeated auctions, PPS1 is bidding at $p_1 = 9.0$ or 10.0, and PPS2 is bidding at $p_2 = 9.0$. In this steady state, the probability that PPS1 bids at $p_1 = 9.0$ is $\frac{a}{a+c}$, and the probability that it bids at $p_1 = 10.0$ is $\frac{c}{a+c}$. The Markov chain analysis suggests that the equilibrium is at $p_1 = 10.0$ even in the repeated auction.

5.3 Replicator Dynamics

The assumed model in this chapter can be analyzed by the discrete-time replicator equation we derived in Chapter 4:

$$x_i(t+1) = x_i(t) + \beta x_i(t)\,(\pi_i - \hat{\pi})$$

Starting from the uniform distribution, after 2000 auctions the probabilities that PPS1 and PPS2 use the various strategies are given in the table below:

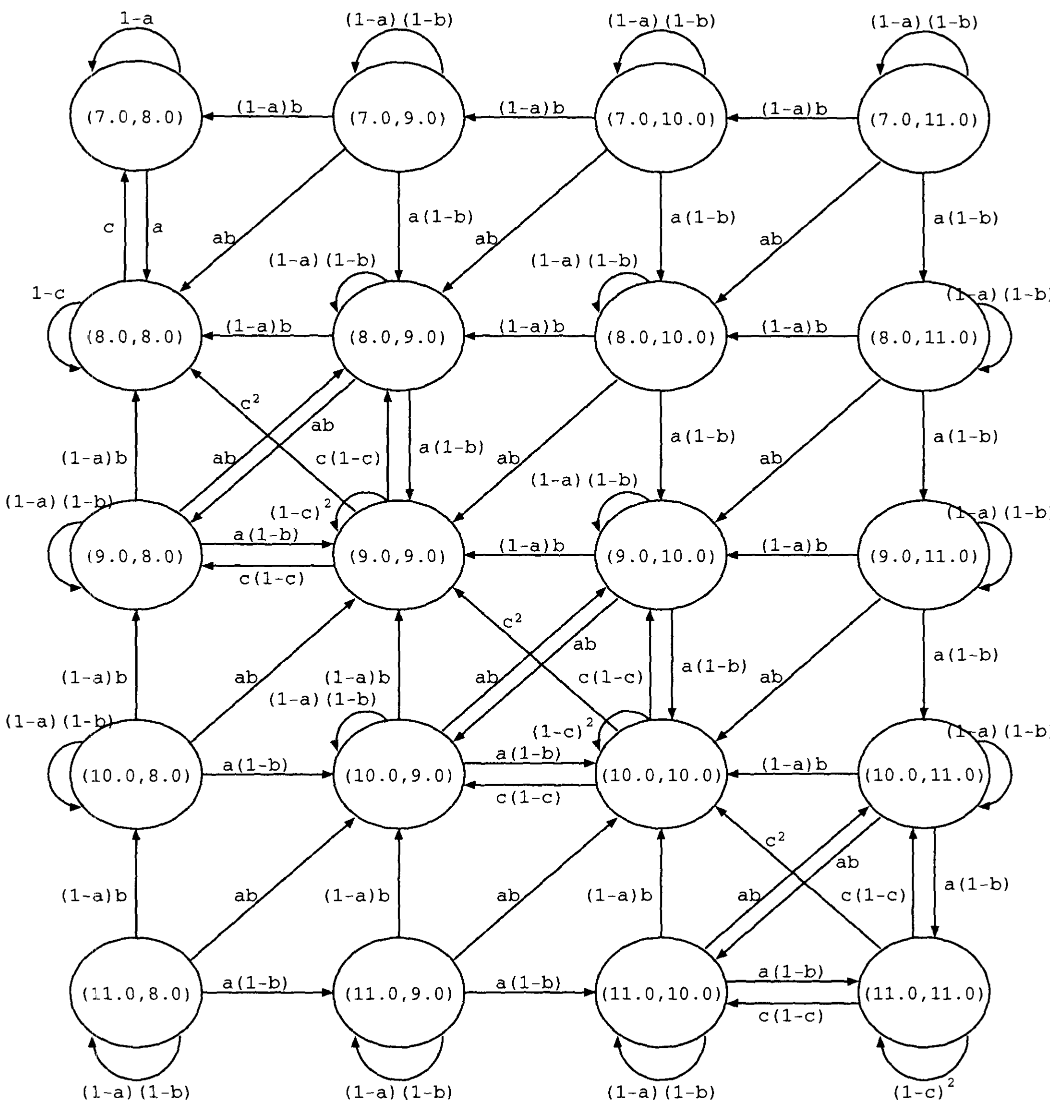

Figure 5-3: Markov chain for the first round of the auction.

Strategy	6.0	7.0	8.0	9.0	10.0
PPS1	0.0000	0.0000	0.0000	0.0000	1.0000
PPS2	-	0.0022	0.0039	0.0112	0.9827

During 2000 auctions, PPS1 shifts from the uniformly distributed mixed strategy to the pure strategy $p_1 = 10.0$. Thus, the replicator dynamics also suggests that the equilibrium exists at $p_1 = 10.0$ in the repeated auction. Note that the result in the table is symmetrical to the result in Section 4.1 due to the symmetrical property of the payoff matrices.

5.4 Summary

Through this chapter we showed that the first round of the auction could be analyzed in the same way as the second round of the auction. In the Nash equilibrium for the non-repeated auction, the player with the higher cost wins the auction with a bid price equal to the cost of the other player. This Nash equilibrium is the same as the Nash equilibrium for the second round of the auction. For the repeated auction, by both the Markov chain analysis and the replicator analysis, we also showed the equilibrium existed at the strategy which was equal to the Nash equilibrium for the non-repeated auction. The consistency of the results for the first round and the second round of the auction comes from the symmetric property of the payoff matrices and the Markov chains.

Chapter 6

Agent-Based Simulation

6.1 Trade Network Game

The Trade Network Game (TNG) is a C++ platform for agent-based simulation [9]. In the TNG, autonomous agents choose their partner-agents, and partner-agents accept and reject offers from them. They update their payoffs through trades, and evolve their trade strategies according to their payoffs.

6.2 Modified Trade Network Game

The TNG has been modified in this thesis to simulate the simple form of the electricity auction in Japan. In the modified TNG, or MTNG, traders consisting of buyers and sellers follow the procedure below to trade electricity in the auction:

1. A buyer announces its intent to buy a particular amount of electricity at prices up to a maximum acceptable value.

2. Sellers bid to sell their electricity to the buyer, announcing amount and bid price.

3. The buyer accepts offers from the sellers according to their bid prices.

4. Sellers update their payoff of the strategy they used in the auction.

In the conventional TNG, when the buyer accepts offers from the sellers according to its preference for the sellers, the preference is decided by the expected payoff derived from past trades. In contrast, in the MTNG, the bid price is the only element of preference for the buyer, and the history of previous trades doesn't affect the preference of the buyer.

Applying the procedure of the MTNG, the assumed model in previous chapters is simulated as follows. At first, PPS0 announces its intent to buy 100 kWh of electricity if it is cheaper than 12 Yen/kWh. Then, PPS1 and PPS2 bid to sell their electricity to PPS0. They both have sufficient generating capacity to supply 100 kWh of electricity to PPS0. After both companies bid to PPS0, the bid with lower price will be accepted by PPS0, and the bid with higher price will be rejected by PPS0. If their bid is at the same price, they each supply 50 kWh of electricity to PPS0. The cost to supply electricity to PPS0 is 5 Yen/kWh for PPS1 and 7 Yen/kWh for PPS2, respectively.

From now on, we shall use the term *reward* to denote the profit in the last auction, and the *payoff* is updated according to the reward they received through all past auctions. Let R denote the reward the trader receives. Since the bid price is not a function of amount of electricity, the reward for traders is calculated as follows:

When the trader wins,

$$R = (\textit{bid price} - \textit{cost})\,[\text{Yen/kWh}] \times 100\,[\text{kWh}]$$

When the trader loses,

$$R = 0\,[\text{Yen}]$$

When the trader ties,

$$R = (\textit{bid price} - \textit{cost})\,[\text{Yen/kWh}] \times 50\,[\text{kWh}]$$

The evolution of traders is the important feature of the TNG. In the TNG, traders cooperate with and defect from their trade-partners strategically, and they update their strategy by evolution. In the MTNG, traders always cooperate with their trade-partners. Their strategy is only to choose the bid price in order to maximize their reward, and the choice of the strategy is made according to the distribution of the payoff. Thus, every update of their payoff according to the reward is equivalent to the evolution in the TNG.

6.3 Update of the Payoff

Traders learn which strategy is best in order to maximize their reward through repeated auctions. From each auction, traders receive their reward, and the payoff of the strategy used in the auction is updated. This update of the payoff and the strategy selection corresponds to the learning process for traders, and the distribution of the payoff over their strategies affects the choice of the strategy.

Since the learning process of traders can affect the dynamics of the repeated auction, we have implemented three kinds of learning algorithms in the MTNG in order to investigate the effect of learning algorithms of traders. For each learning algorithm, after the i-th strategy is used in the auction for the k-th time, the payoff for the i-th strategy is updated as follows:

1. Average Learning
$$\pi_i \longleftarrow \frac{R + (k-1)\pi_i}{k}$$
2. Exponential Learning
$$\pi_i \longleftarrow \frac{R + \rho(k-1)\pi_i}{k}$$
3. Q-Learning
$$\pi_i \longleftarrow \eta R + (1-\eta)\,\pi_i$$

where

π_i	:	payoff for the i-th strategy
		$\pi_i = 0$ for $\forall i$ before the i-th strategy is used for the first time
ρ	:	coefficient for the exponential learning algorithm (0.8)
η	:	coefficient for the Q-learning algorithm (0.01)

In the average learning algorithm, the payoff is updated to be equal to the average reward the trader has received when it used the i-th strategy. In the exponential learning algorithm, the payoff is again updated to show the average reward, as the payoff in the average learning algorithm, but past rewards are exponentially weighted by the coefficient ρ. This coefficient ρ can take the value from 0 to 1; when $\rho = 0$, only the newest reward is considered, when $\rho = 1$, the rewards at all auctions are given the same emphasis, which is the same as the average learning algorithm. We use $\rho = 0.8$ in this thesis. Thus, the rewards at the old auctions diminish by the coefficient $\rho = 0.8$. When the auction environment is changing, this emphasis on recent results can be an advantage for traders to adapt quickly. However, in this thesis, since the auction environment is fixed throughout 1500 iterations, it makes a trader forget its past rewards more quickly. In the Q-learning algorithm, the newest reward and accumulated payoff are taken into account in the proportion of η to $1 - \eta$, where η can take values from 0 to 1, and $\eta = 0.01$ is used in this thesis.

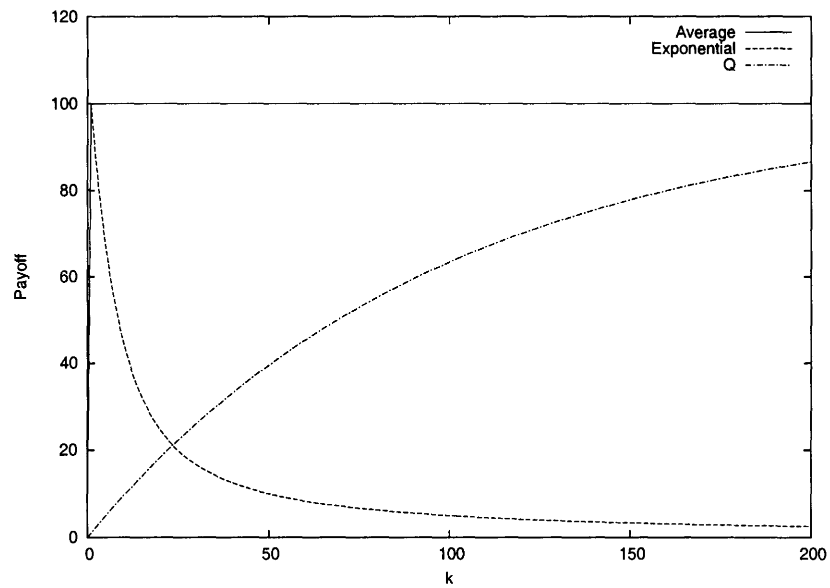

Figure 6-1: Comparison between three learning algorithms.

The differences of the three learning algorithms are characterized in Figure 6-1. Assuming that every time a trader uses some successful strategy, it receives a reward of 100, Figure 6-1 is derived from the three learning algorithms. The horizontal axis k of the figure represents how many times the strategy is used by the trader. Under the average learning algorithm, the payoff is kept equal to 100 regardless of k. Under the exponential learning algorithm, the payoff decreases from 100 with increasing k. Thus, when the trader's strategy selection starts to converge, the exponential learning algorithm slows down the convergence. This way, the trader can try more strategies than in the average learning algorithm. However, after the same strategy is used about fifty times, the payoff of the strategy doesn't decrease very fast with increasing k, and the trader's strategy selection can converge to the strategy. In contrast, under the Q-learning algorithm, the payoff is equal to 1 when the strategy is first used, and asymptotically come close to 100 with increasing k. Since the payoff is small at first, the convergence speed is relatively slow until the trader's strategy selection starts to converge. Therefore, the trader can try more strategies before that than in the average learning algorithm.

6.4 Strategy Selection

In the MTNG, the bid price that traders can choose as their strategy is discrete. The bid price, p, can be

$$p = cost \times (1 + 0.02 \times i), \quad i = 1, 2, 3, \cdots$$

and p has to be smaller than maximum acceptable price.

Traders decide which strategy to use according to the distribution of the payoffs over their strategies. Since the algorithm for the strategy selection can also affect the dynamics of the repeated auction, three types of algorithms for the strategy selection are adopted in the MTNG.

Let $x_i(t)$ denote the probability that the i-th strategy is chosen by the trader at time t, then $x_i(t)$ is calculated as follows for each strategy selection algorithm $(i = 1, 2, \cdots, n)$.

1. Replicator Selection

$$x_i(t+1) = x_i(t) + \beta x_i(t) \cdot (\pi_i - \hat{\pi})$$

2. ϵ-Greedy Selection

$$x_i(t+1) = \begin{cases} (1-\epsilon) + \frac{\epsilon}{n} & \text{, if } i\text{-th strategy is the strategy with the best payoff} \\ & \quad \text{at time } t \\ \frac{\epsilon}{n} & \text{, otherwise} \end{cases}$$

3. Payoff-Weighted Selection

$$x_i(t+1) = \frac{\pi_i}{\sum_{j=1}^{n} \pi_j}$$

where

$\hat{\pi}$: average payoff over all strategies
($= \pi_1 x_1 + \pi_2 x_2 + \cdots + \pi_n x_n$, for n strategies)

β : coefficient for the replicator equation (0.01)

ϵ : coefficient for the ϵ-greedy selection (0.5)

In the ϵ-greedy selection, traders choose the strategy randomly with a probability of ϵ, anticipating that they may find a better strategy. On the other hand, they use the best strategy so far with a probability of $1-\epsilon$. In the payoff-weighted selection, the probability that traders choose the i-th strategy is proportional to π_i.

At the first auction, the replicator selection and the ϵ-greedy selection choose one strategy over all strategies with equal probability. In contrast, since the payoff-weighted selection requires an initial payoff distribution, the ϵ-greedy selection is used for the first 500 iterations, and the payoff-weighted selection is used for the last 1000 iterations.

6.5 Simulation Results

6.5.1 Results with the Average Learning Algorithm

In the assumed model of the MTNG, when $5.0 < p_1 \leq 7.0$, PPS1 always wins the auction and receives the reward, which is linear in p_1. Thus, PPS1's payoff at $5.0 < p_1 \leq 7.0$ is linear under the average learning algorithm. When $p_1 > 7.0$, PPS1 doesn't always win but receives a larger reward when it wins.

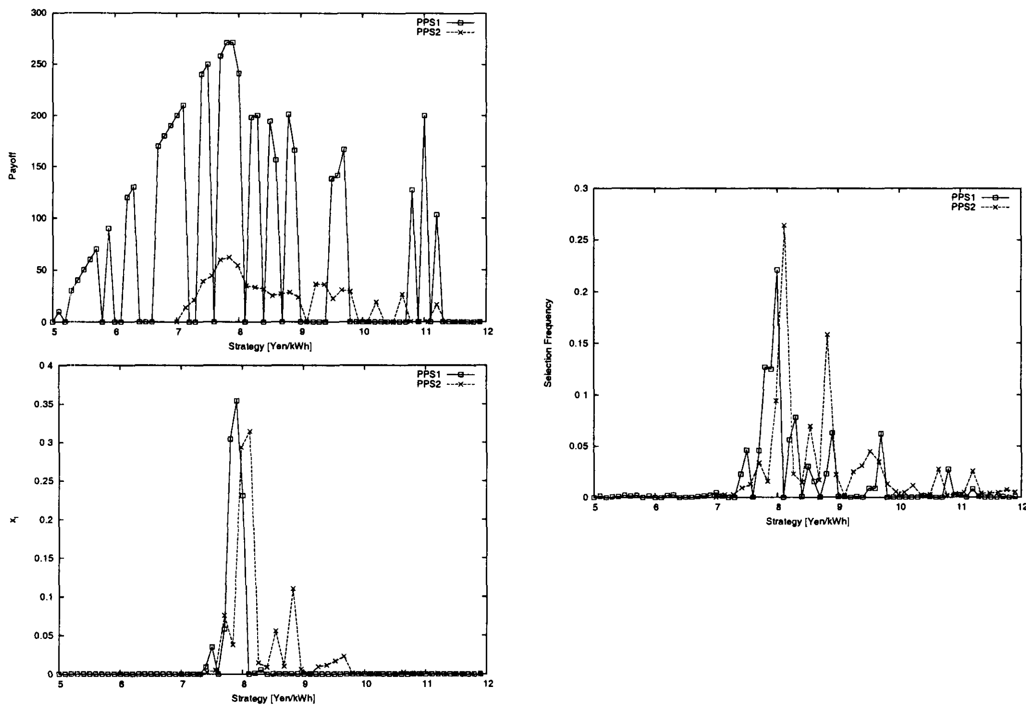

Figure 6-2: Results with average learning algorithm and replicator selection.

Figures 6-2, 6-3 and 6-4 show the results with the average learning algorithm. The top-left graph of each figure shows the payoff after 1500 auctions. For all strategy selections, when $7.0 < p_1 < 9.0$, PPS1 earns the average payoff, which is about 200 - 300, and the average payoff decreases with increasing p_1. Thus, PPS1 and PPS2 don't find Pareto efficient strategies as profitable strategies for them. In the replicator selection, PPS1 has zero payoff at some strategies even at $5.0 < p_1 \leq 7.0$, which suggests that PPS1 didn't try all its strategies in 1500 auctions. By lowering

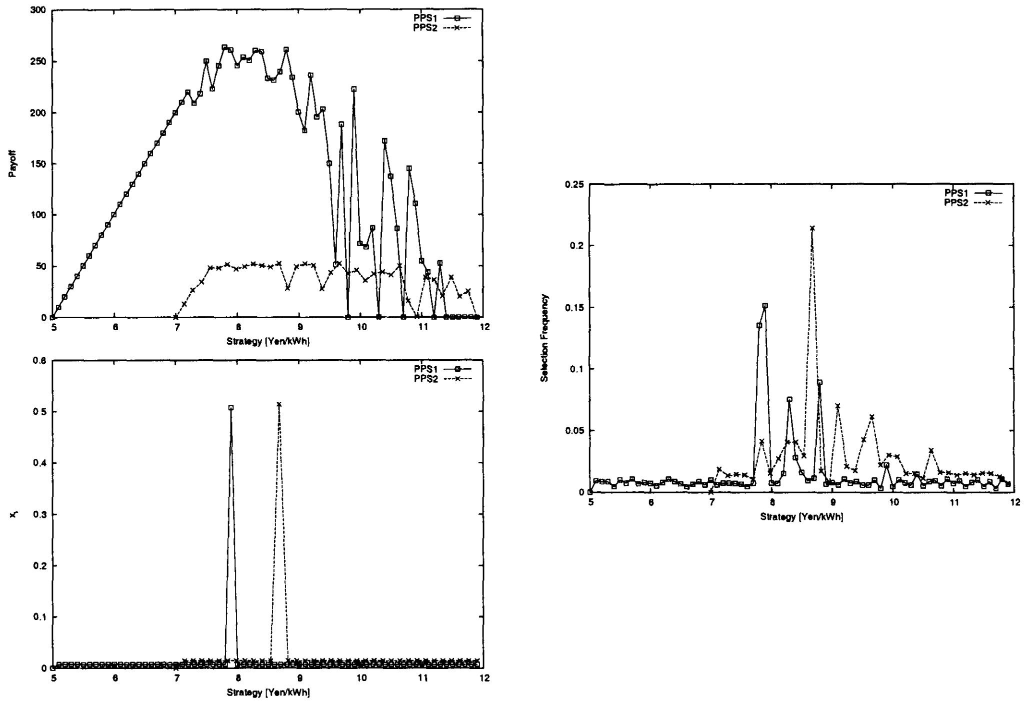

Figure 6-3: Results with average learning algorithm and ϵ-greedy selection.

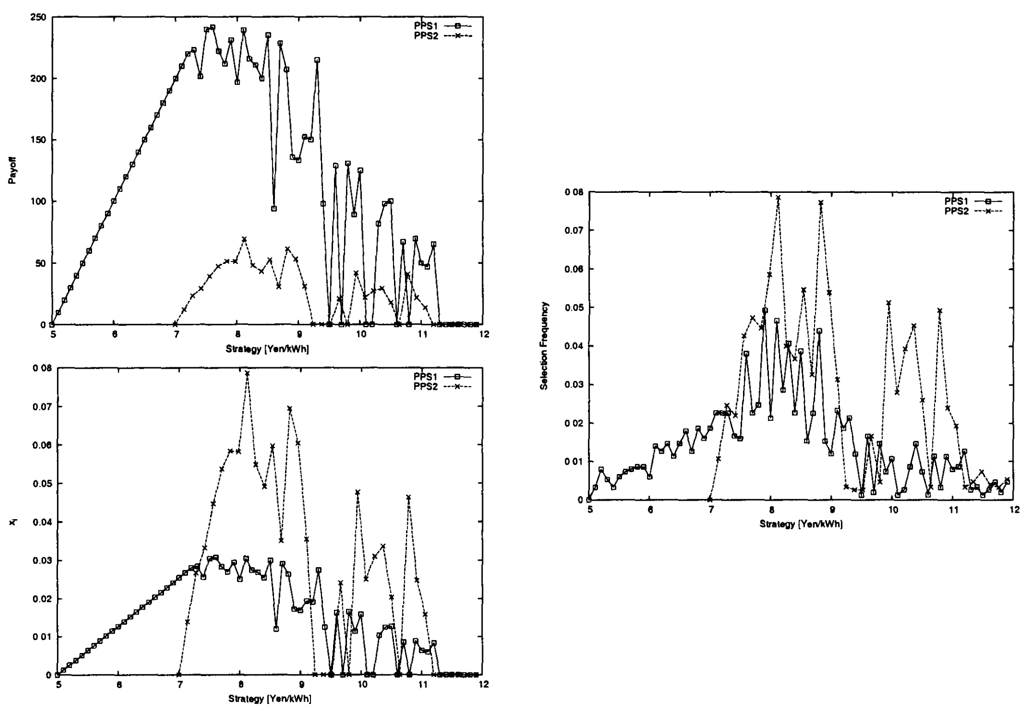

Figure 6-4: Results with average learning algorithm and payoff-weighted selection.

the value of β from 0.01, we can make PPS1 explore more strategies at the cost of the convergence speed.

The bottom-left graph of each figure shows the probability that each strategy is used after 1500 auctions, and the right graph of each figure shows the frequency with which each strategy has been chosen in 1500 auctions. From these graphs, we can see the trader's strategies converge relatively quickly under the replicator selection and the ϵ-greedy selection. PPS1's strategy converges to $7.0 \leq p_1 < 9.0$, and it is always lower than the strategy to which p_2 converges. Note that $p_1 = 7.0$ is the equilibrium previously derived from our Markov chain analysis and replicator dynamics analysis.

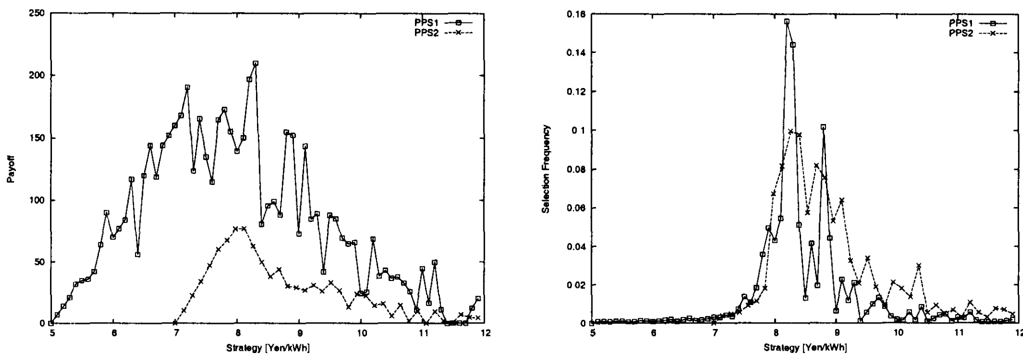

Figure 6-5: Average results of ten executions with average learning algorithm and replicator selection.

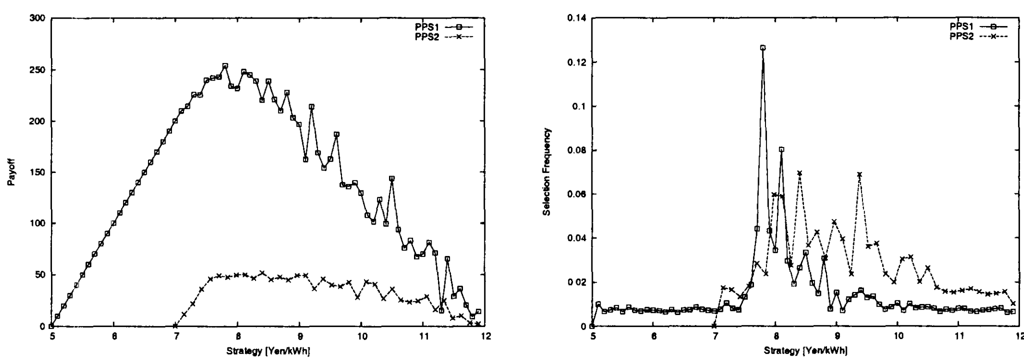

Figure 6-6: Average results of ten executions with average learning algorithm and ϵ-greedy selection.

In the MTNG, since traders choose one strategy according to the probabilities $x_i(t)$ at time t, randomness exists in their strategy selection. Thus, the average results of

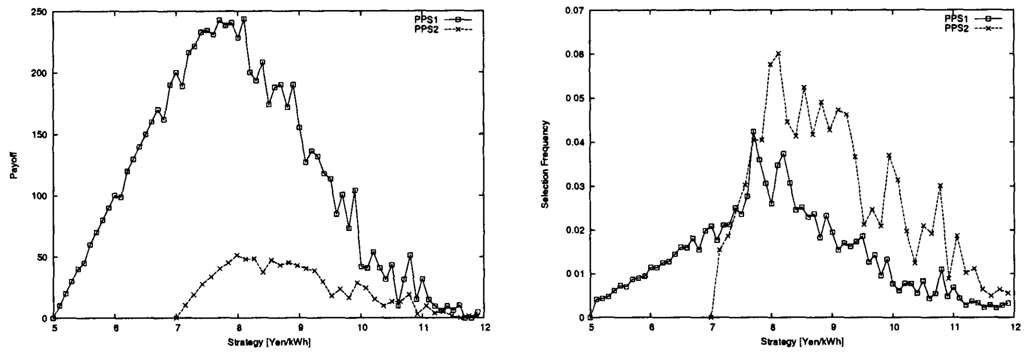

Figure 6-7: Average results of ten executions with average learning algorithm and payoff-weighted selection.

ten executions with all strategy selections are shown in Figure 6-5, 6-6 and 6-7. In Figure 6-5, we see PPS1 uses all strategies between 5.0 to 7.0 in 1500 auctions even with the replicator selection. We can also see that the characteristics discussed for the results of one execution hold for the average results of ten executions.

6.5.2 Results with the Exponential Learning Algorithm

Figures 6-8, 6-9 and 6-10 show the results with the exponential learning algorithm. Because of the characteristics of the exponential learning algorithm that are discussed earlier, PPS1 chooses the strategies $5.0 < p_1 < 7.0$ more often than in the average learning algorithm, which don't bring better rewards than the strategy $p_1 = 7.0$. Also, PPS1 chooses the strategies $9.0 \leq p_1 < 12.0$ more often than in the average learning algorithm, which don't bring better rewards than the strategies $7.0 \leq p_1 < 9.0$, considering PPS1 more likely to lose to PPS2 when it uses the strategies $9.0 \leq p_1 < 12.0$.

Though we can find these difficulties for PPS1's strategy selection to converge to $7.0 \leq p_1 < 9.0$, the strategies $7.0 \leq p_1 < 9.0$ are most frequently used by PPS1 compared with other strategies in the same way as in the average learning algorithm. The higher payoffs of PPS1 at the strategies $5.0 < p_1 < 7.0$ show that the strategies $5.0 < p_1 < 7.0$ are not frequently used by PPS1 and that the exponential learning algorithm has the effect of making PPS1 use these strategies by giving them higher

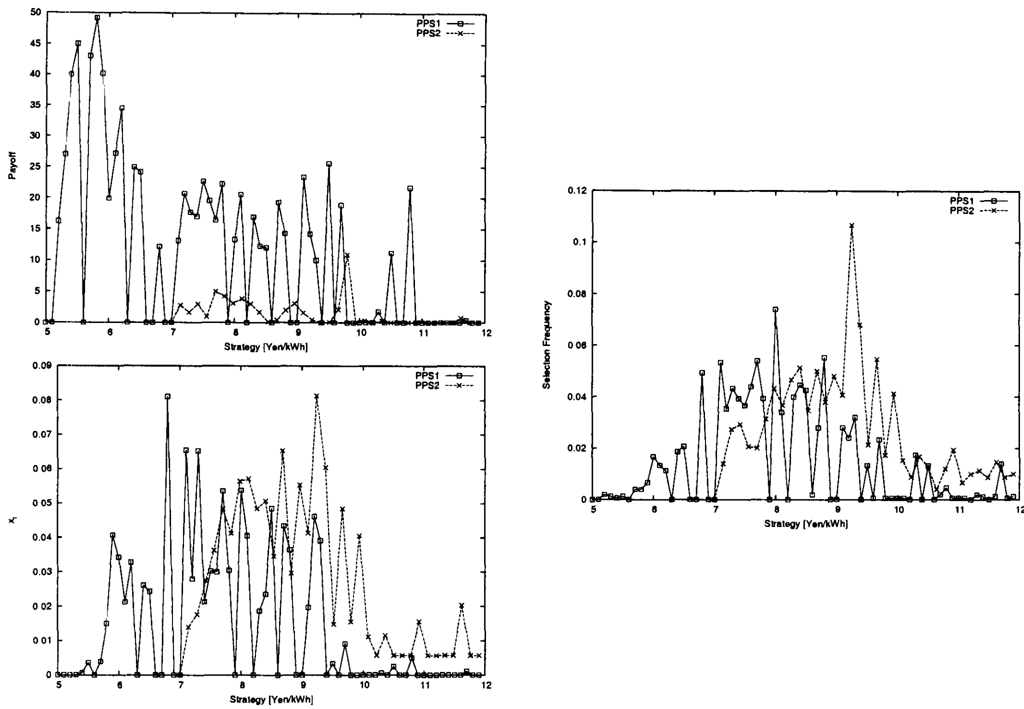

Figure 6-8: Results with exponential learning algorithm and replicator selection.

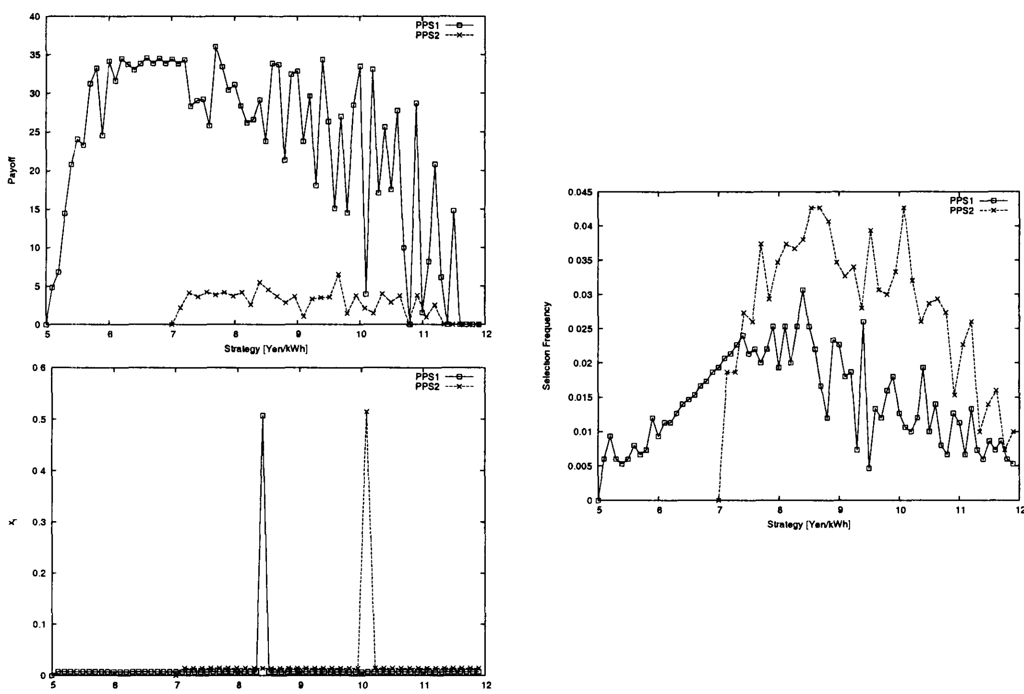

Figure 6-9: Results with exponential learning algorithm and ϵ-greedy selection.

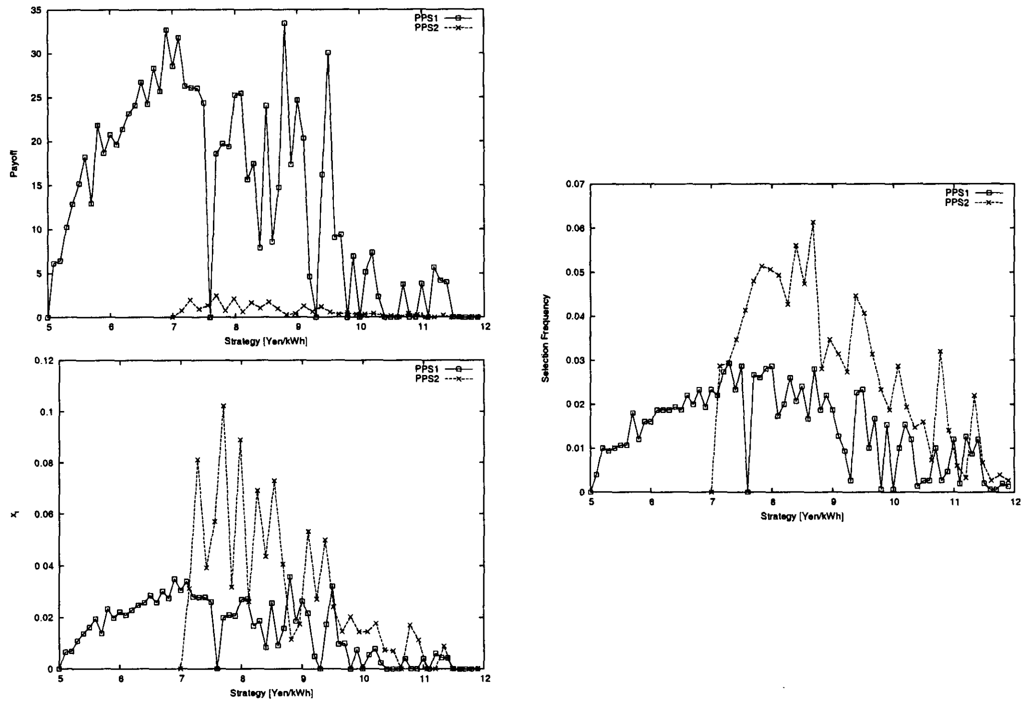

Figure 6-10: Results with exponential learning algorithm and payoff-weighted selection.

payoffs. Because of these difficulties, we don't see the obvious difference in the convergence speed between the three strategy selections that we saw under the average learning algorithm.

The average results of ten executions with all strategy selections are shown in Figures 6-11, 6-12 and 6-13. The characteristics discussed for the results of one execution hold for the average results of ten executions.

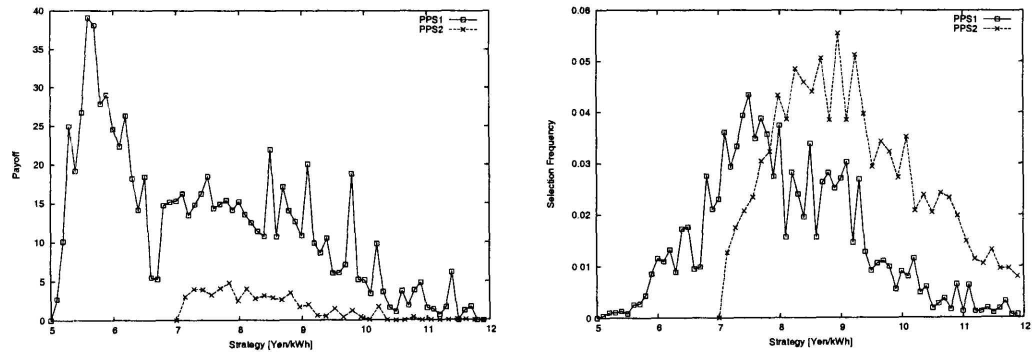

Figure 6-11: Average results of ten executions with exponential learning algorithm and replicator selection.

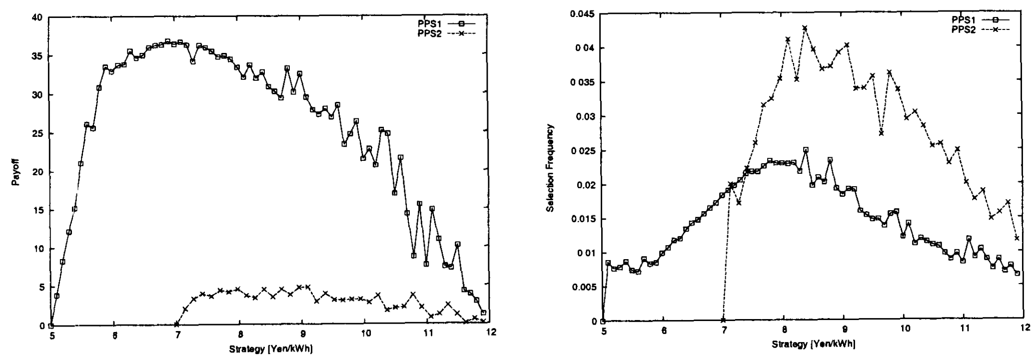

Figure 6-12: Average results of ten executions with exponential learning algorithm and ϵ-greedy selection.

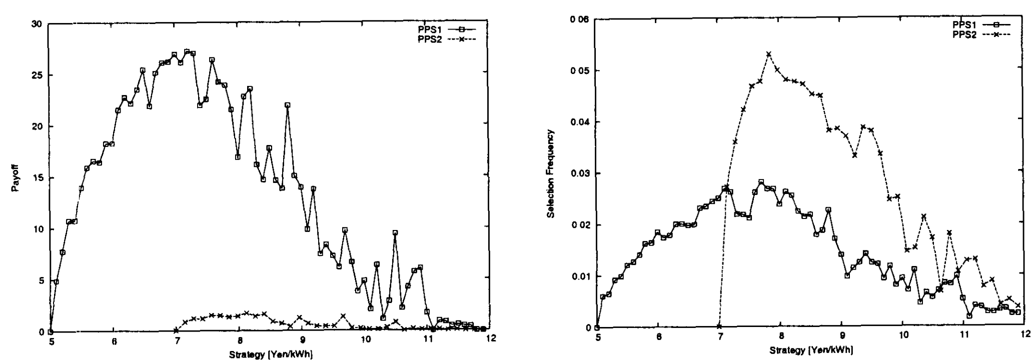

Figure 6-13: Average results of ten executions with exponential learning algorithm and payoff-weighted selection.

6.5.3 Results with the Q-Learning Algorithm

Figures 6-14, 6-15 and 6-16 show the results with the Q-learning algorithm. Because of the characteristics of the Q-learning algorithm, PPS1 and PPS2 try more strategies than in other learning algorithms. In Figure 6-14, we can see all strategies between 5.0 to 7.0 are used at least once by PPS1 even in the replicator selection. At the same time, since a trader receives a higher payoff as it uses the same strategy repeatedly, we can see trader's strategies converge relatively quickly under the replicator selection and the ϵ-greedy selection compared to the payoff-weighted selection, as we see in the average learning algorithm.

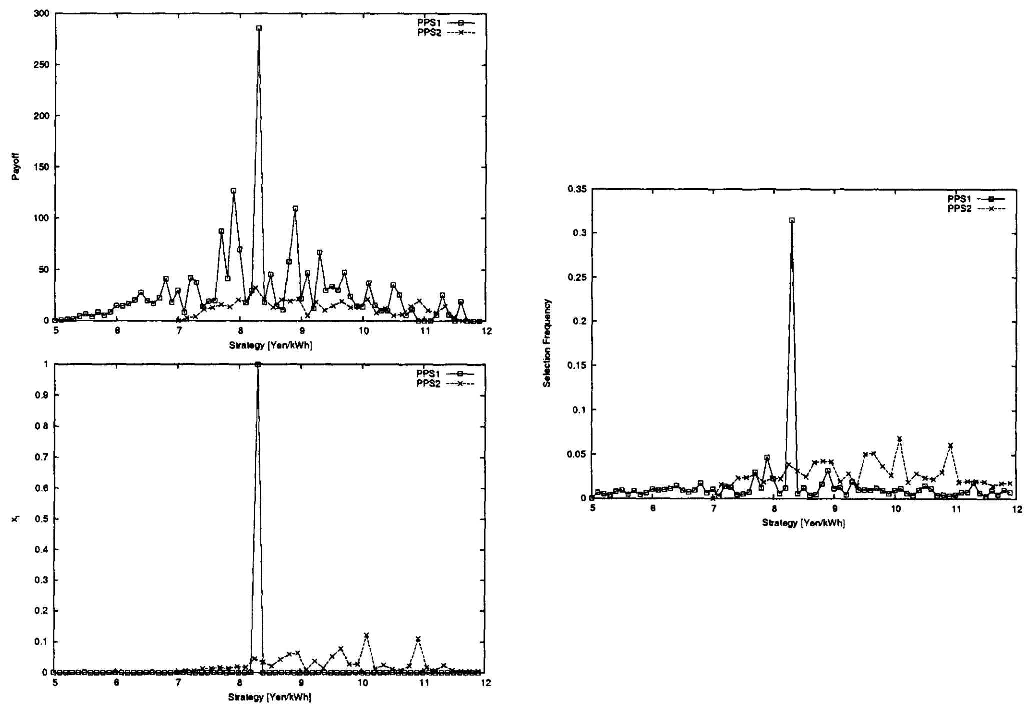

Figure 6-14: Results with Q-learning algorithm and replicator selection.

The average results of ten executions with all strategy selections are shown in Figure 6-17, 6-18 and 6-19. Once the convergence to a strategy is started in the repeated auction, the convergence will be accelerated in the Q-learning algorithm. Due to this characteristic, we see that trader's strategy selection converges to a different strategy in every execution.

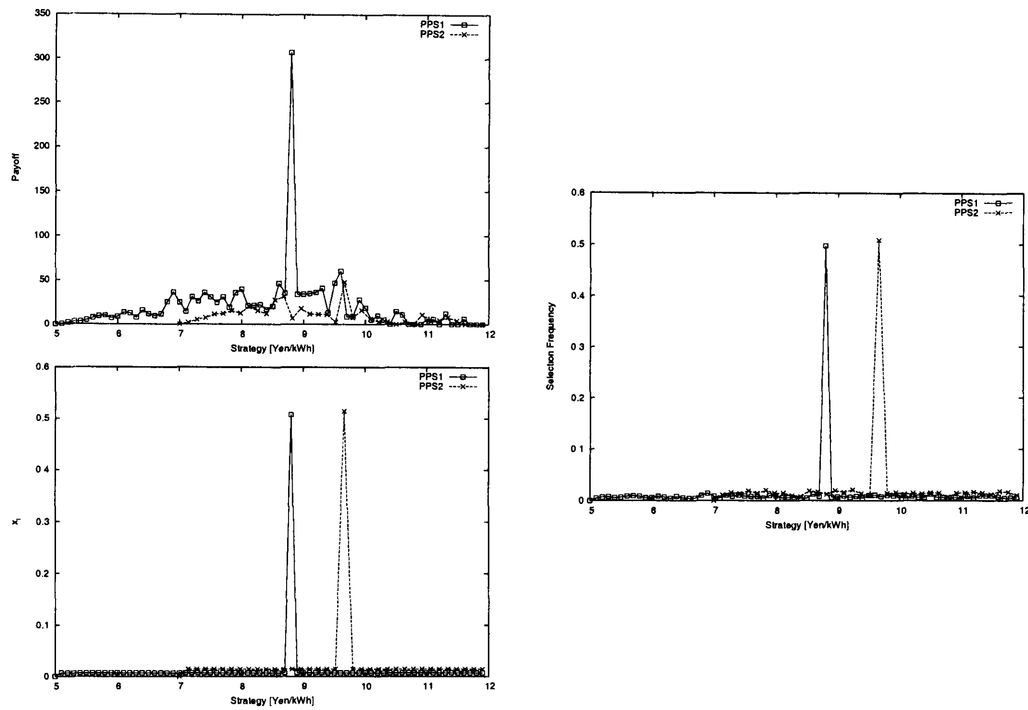

Figure 6-15: Results with Q-learning algorithm and ϵ-greedy selection.

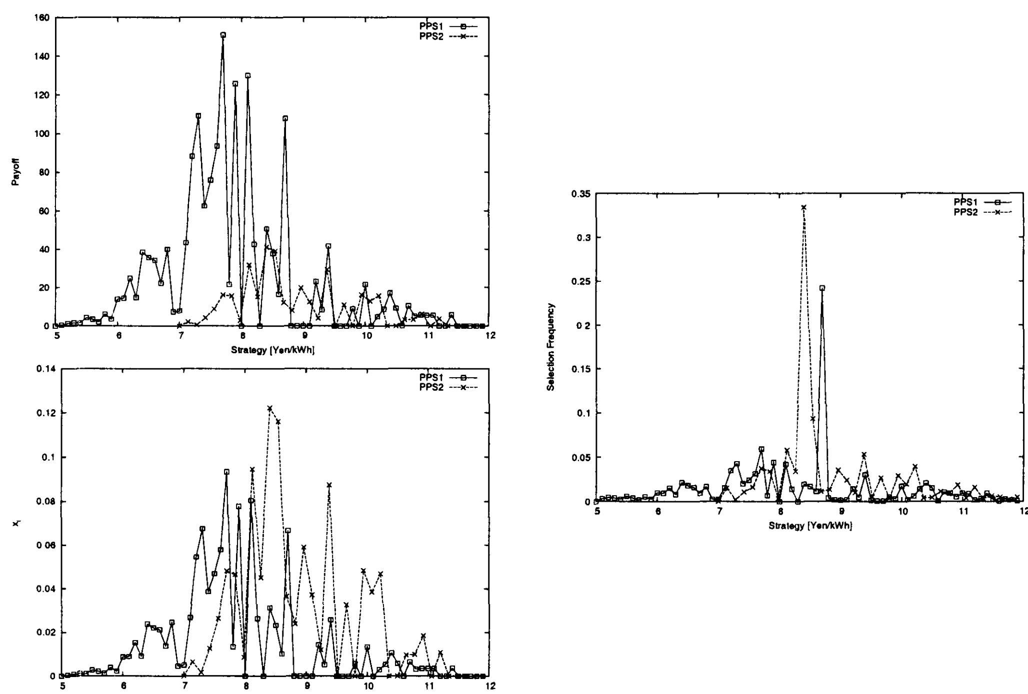

Figure 6-16: Results with Q-learning algorithm and payoff-weighted selection.

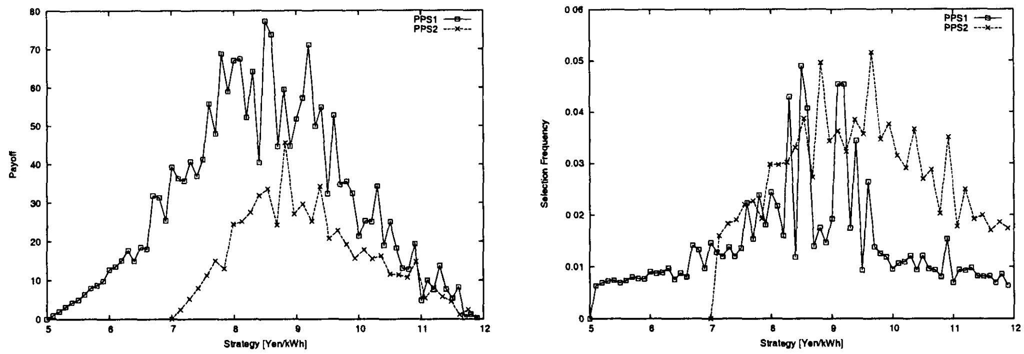

Figure 6-17: Average results of ten executions with Q-learning algorithm and replicator selection.

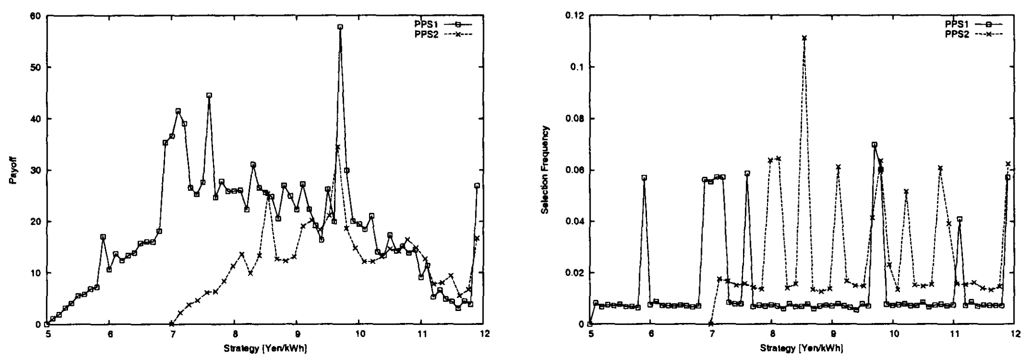

Figure 6-18: Average results of ten executions with Q-learning algorithm and ϵ-greedy selection.

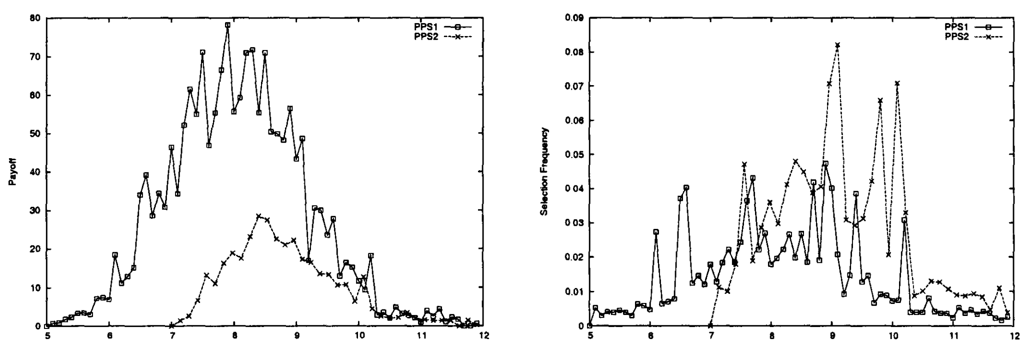

Figure 6-19: Average results of ten executions with Q-learning algorithm and payoff-weighted selection.

Chapter 7

Discussion and Conclusion

7.1 Penalty for Deviating from Tacit Collusion

Though some of the literature has pointed out the potential for tacit collusion in electricity markets [5] [10], our results show that the market mechanism promotes more competition than collusion. One of the reasons derives from the difference in the bidding curve.

In most electricity markets in the world, participants specify in their bid a set of points defined by quantity and price of electricity. In some markets such as Nord Pool, these points are connected by straight lines to form a bidding curve as shown in Figure 7-1. In contrast, in other markets such as JPEX, these points are connected by horizontal and vertical lines to form a bidding curve as shown in Figure 7-2. We call this bidding curve a stepwise bidding curve in this thesis. In Figure 7-1 and Figure 7-2, round circles mark a set of points specified in seller bids by PPS1 and PPS2.

Now, we show that the bidding curves in Figure 7-1 penalize participants when one of them deviates from tacit collusion. First, we consider the simple bidding curves shown in Figure 7-3. Suppose that under some collusion PPS1 and PPS2 bid with bidding curves $q = a_1p$ and $q = a_2p$, respectively ($a_2 > a_1$). Or more generally, suppose that PPS1 and PPS2 can set more competitive bidding curves than that but don't pursue competition eagerly in Figure 7-3. Here, q and p stand for quantity and price, and a_1 and a_2 are coefficients decided by the specification in

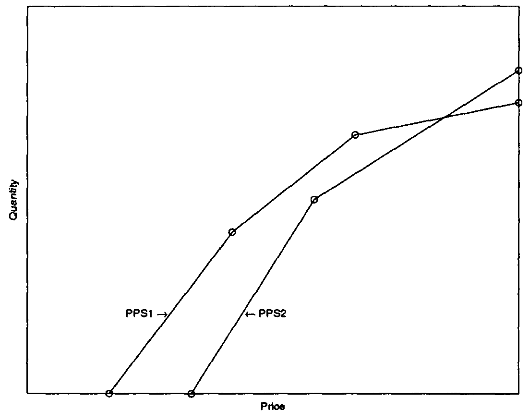

Figure 7-1: Bidding curve (Nord Pool type).

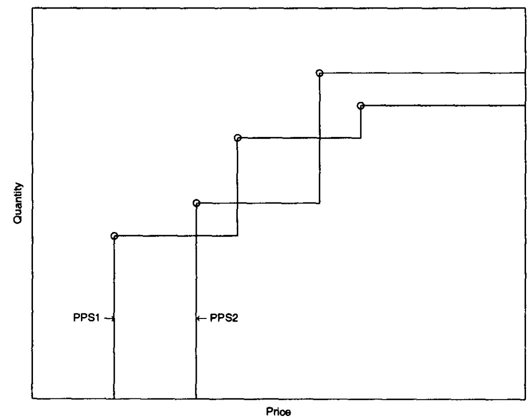

Figure 7-2: Bidding curve (JPEX type).

their bids. Assuming inelastic demand d, the market clearing price can be calculated as $p^* = \dfrac{d}{a_1 + a_2}$. The respective quantities of electricity PPS1 and PPS2 sell at the market is $q_1^* = a_1 p^*$ and $q_2^* = a_2 p^*$.

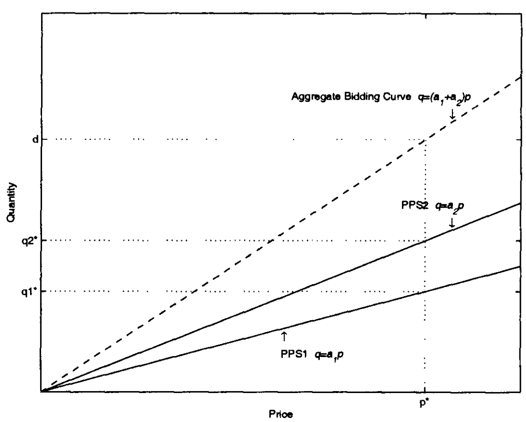

Figure 7-3: Market under collusion.

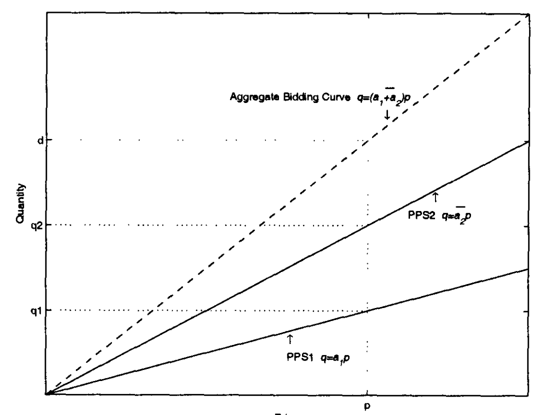

Figure 7-4: Deviation from Figure 7-3.

Next, suppose that PPS2 sets a more competitive bidding curve $q = \overline{a_2}p$ ($\overline{a_2} > a_2$) as shown in Figure 7-4 in order to increase its profit. For the same inelastic demand d, the new market clearing price will be $p = \dfrac{d}{a_1 + \overline{a_2}}$ ($< p^*$). At this price, PPS1 and PPS2 sell $q_1 = a_1 p$ and $q_2 = \overline{a_2}p$ of electricity, respectively.

Let u_1^* and u_1 be the respective payoffs PPS1 receives at the market in Figure 7-3 and Figure 7-4. Also, let c_1 denote the cost to PPS1 of producing a unit of electricity.

Comparing these two payoffs,

$$\begin{aligned} u_1 - u_1^* &= q_1(p - c_1) - q_1^*(p^* - c_1) \\ &= a_1 p(p - c_1) - a_1 p^*(p^* - c_1) \\ &= a_1(p^2 - p^{*2}) - a_1 c_1(p - p^*) \\ &= a_1(p - p^*)(p + p^* - c_1) \end{aligned}$$

Since $c_1 < p < p^*$, we get $u_1 - u_1^* < 0$, which suggests that the payoff to PPS1 decreases by the deviation of PPS2.

We next consider the payoff to PPS2. Defining u_2^*, u_2, and c_2 in the same way as u_1^*, u_1, and c_1,

$$\begin{aligned} u_2 - u_2^* &= q_2(p - c_2) - q_2^*(p^* - c_2) \\ &= \overline{a_2} p(p - c_2) - a_2 p^*(p^* - c_2) \\ &= \overline{a_2} p^2 - a_2 p^{*2} + c_2(a_2 p^* - \overline{a_2} p) \\ &= d^2 \left\{ \frac{\overline{a_2}}{(a_1 + \overline{a_2})^2} - \frac{a_2}{(a_1 + a_2)^2} \right\} + dc_2 \left\{ \frac{a_2}{a_1 + a_2} - \frac{\overline{a_2}}{a_1 + \overline{a_2}} \right\} \\ &= \frac{d^2(\overline{a_2} - a_2)(a_1^2 - a_2 \overline{a_2})}{(a_1 + \overline{a_2})^2 (a_1 + a_2)^2} + \frac{dc_2 a_1(a_2 - \overline{a_2})}{(a_1 + \overline{a_2})(a_1 + a_2)} \end{aligned}$$

We can see both terms are negative because $a_1 < a_2 < \overline{a_2}$. Thus, $u_2 - u_2^* < 0$, so the payoff of PPS2 also decreases by its own deviation.

From this observation, we see that the deviation of PPS2 penalizes both PPS1 and PPS2, regardless of demand level. Here, we showed the penalty rule for the simple bidding curves in Figure 7-3. The same rule often holds for the more general bidding curves shown in Figure 7-1. For these general bidding curves, we can carry out a similar analysis for each block of a demand level. This penalization of deviations from collusion plays an imperative role in maintaining collusion in some simulations of electricity auctions [10].

Next, we consider stepwise bidding curves and their ability to penalize a deviation from collusion. Figure 7-5 shows an example of bids using stepwise bidding curves.

The situation is similar to what we analyzed by Figure 7-3 and Figure 7-4, and we assume this market is under some collusion. For some inelastic demand d, we have market clearing price p^* as shown in Figure 7-5. At this price p^*, PPS1 and PPS2 sell q_1 and q_2 of electricity, respectively.

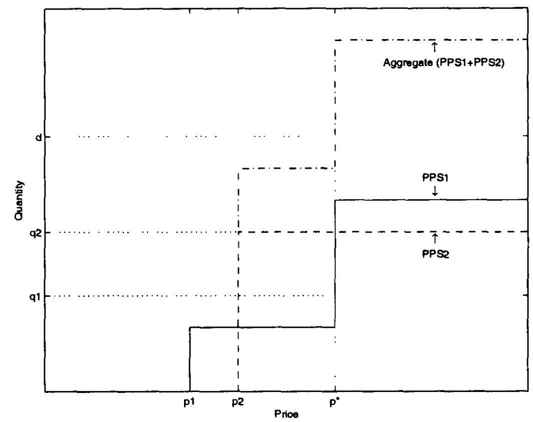

Figure 7-5: Stepwise curves with high demand.

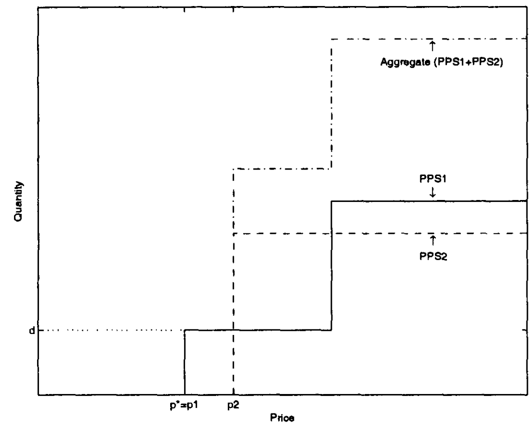

Figure 7-6: Stepwise curves with low demand.

Now, we consider a deviation from Figure 7-5. Suppose that PPS2 tries to set a more competitive bidding curve in order to increase its profit. PPS2 has two choices, either move the line $p = p_2$ left or move the line $q = q_2$ up. When PPS2 moves the line $p = p_2$ left, there will be no effect to the market. Both PPS1 and PPS2 will not be penalized, and they will not gain anything. When PPS2 moves the line $q = q_2$ up a little, the quantity of electricity PPS2 sells at the market will increase, while the quantity of electricity PPS1 sells at the market will decrease. Thus, these bidding curves don't penalize PPS2 when PPS2 deviates from collusion. However if PPS2 moves the line $q = q_2$ up to a large extent, for example above the top of the bidding curve of PPS1 in this figure, the market clearing price p^* will go down to p_2, and PPS2 will be penalized in many cases.

Figure 7-6 is a low demand case of Figure 7-5, and the bidding curves in the figure are same as those in Figure 7-5. The market clearing price p^* is as low as p_1, and PPS1 covers all the demand $d = q_1$. If PPS2 wants to set a more competitive bidding curve at this market, it has to move the line $p = p_2$ left. When the deviation is small, there will be no effect on the market. When PPS2 sets p_2 lower than p_1, however, it will take over all the demand d from PPS1, and there is no penalty for PPS2 in this process.

In the model assumed in this thesis, a buyer announces its demand before sellers bid to the market. Since sellers bid knowing the demand level, bidding curves have the shape shown in Figure 7-7, and the situation always realizes a low demand market with stepwise bidding curves. Therefore, bidding curves in our assumed model don't penalize deviation from collusion, and it is one of the reasons that we couldn't observe tacit collusion in this thesis.

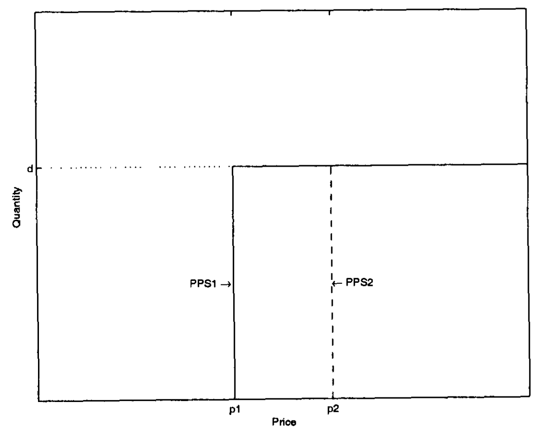

Figure 7-7: Bidding curves in our assumed model.

7.2 Comparison of the Methods Used to Find an Equilibrium

We have used the iterated elimination of dominated strategies, Markov chain analysis, replicator analysis, and agent-based simulation to find an equilibrium for the auction. These methods have different advantages and limitations. In this section, we discuss these limitations and their effects on the results.

Most importantly, in the iterated elimination of dominated strategies, we find Nash equilibria of a non-repeated auction. This is a static analysis, and we cannot see a dynamic characteristic through this analysis.

Also, we cannot consider continuous strategy space. The number of strategies has to be finite to reach Nash equilibria by a finite number of eliminations. However, no matter how small we divide a continuous strategy space into discrete strategy spaces, we obtain the same result. Therefore, we don't lose generality by the discretization.

Additionally, we cannot find a mixed-strategy Nash equilibrium by the iterated elimination of dominated strategies. This method leads us only to pure-strategy Nash equilibria. In order to find a mixed-strategy Nash equilibrium for non-repeated auction, we can use the Fundamental Theorem.

Second, in Markov chain analysis, players can carry over information only from the last game. This is also true in replicator analysis. Thus, in these methods, players don't utilize past results beyond the result in the last game. In that sense, the intelligence of players is limited. However, we do see the dynamics of the repeated auction as a sequence of two consecutive auctions.

Again, we cannot consider a mixed-strategy equilibrium by Markov chain analysis. Though it is possible to define a state as a mixed-strategy used in an auction, this increases the number of the states considerably. In our assumed model, PPS1 has five pure-strategies. If we discretize the probability of using these strategies into 5% intervals, PPS1 has $\binom{20+5-1}{5-1} = 10626$ mixed-strategies. In the same way, PPS2 has four pure-strategies, so it will have $\binom{20+4-1}{4-1} = 1771$ mixed-strategies. Thus, if we define a state as a combination of mixed-strategies used by PPS1 and PPS2, we will have 1.9×10^7 states. It will make the analysis intractable.

Also, in Markov chain analysis, we limited a state transition from a state to a state next to it, as shown in Figure 3-1. This limitation is set so that we can see the dynamics of the repeated auction better, and it does not derive from a limitation of Markov chain analysis. Suppose we allow a state transition from a state to a state two intervals away, for example, a state transition from $p_1 = 7.0$ and $p_2 = 10.0$ to $p_1 = 9.0$ and $p_2 = 10.0$ in Figure 3-1. These bold transitions are more likely to happen when we discretize the strategy space into more small intervals. Under this state transition rule, the Markov chain does not have an absorption class as an equilibrium of the repeated auction. When we set the probability of these bold transitions to be low, we will see a similar result. If we set the probability high, however, there can be other interesting results, but we will have to undertake a numerical simulation to analyze the dynamics.

Third, in replicator analysis, we can consider mixed-strategies effectively. After considering mixed-strategies, we have found that under an equilibrium PPS1 uses a pure-strategy equal to the Nash equilibrium in the non-repeated auction.

Replicator analysis is similar to the genetic algorithm in the sense that a subject evolves according to its payoff. Here, the subject represents a strategy. In the genetic algorithm, the evolution is often made by replacing low-payoff subjects with high-payoff subjects, which are called *elite*. In contrast, in replicator analysis, this evolution has to be made in accordance with a replicator equation. A coefficient α and β in a replicator equation corresponds with the rate of the replacement in the genetic algorithm. The coefficient can be determined by trial and error so that we can assure an appropriate evolution speed and the stability of the dynamics.

We used the discrete-time replicator equation derived from the (continuous-time) replicator equation. A set of replicator equations constitutes a set of first-order differential equations and is often solved analytically. If we could solve the replicator equation for our assumed model analytically, we would see the dynamics of the repeated auction more thoroughly. Since in our assumed model the payoff for a player changes according to a mixed-strategy of the other player, however, we cannot solve the replicator equation analytically. Thus, we have to use the discrete-time replicator equation, and it has to be solved numerically.

Last, in agent-based simulation, we can say there are no limitations for the method itself. This approach is therefore desirable, when we implement learning algorithms and strategy selection methods that more closely model decision making processes that auction participants actually follow in the real world. In this thesis, we implemented three learning algorithms: the average learning algorithm, exponential learning algorithm, and Q-learning algorithm. We used two strategy selection methods:

ϵ-greedy selection and payoff-weighted selection. Though these are popular algorithms and methods in reinforcement learning and machine learning, we don't have enough data to confirm that they are similar to actual decision making processes. In our assumed model, we obtained similar results under these various algorithms and methods, but they could affect a simulation result considerably under different models. I hope in future actual decision making processes will be implemented with the cooperation of PPSs.

7.3 Utilization of Additional Information

We have studied the dynamics of repeated games by Markov chain analysis, replicator analysis, and agent-based simulation. In these analyses, however, players have a limitation in their utilization of information obtained from the results in past auctions. The limitation is that a player utilizes only its own results and doesn't utilize the other player's results disclosed when a market settles.

In our model, when PPS1 tries to take advantage of this disclosed information, it can obtain the following two kinds of information about c_2, the cost of PPS2.

1. PPS2 won or tied in the first-round auction with the bid price p_{20}.

   $\Rightarrow c_2 > p_{20}$

2. PPS2 won or tied in the second-round auction with the bid price p_{20}.

   $\Rightarrow c_2 < p_{20}$

Note that p_{20} won't be disclosed when PPS2 loses.

Now, we consider the first information, $c_2 > p_{20}$. If $p_{20} < c_1$, PPS1 doesn't have a way to utilize this information. However, when $p_{20} > c_1$, PPS1 can utilize this information effectively in the second-round auction. Knowing it can always win the second-round auction bidding with the bid price $p_1 \leq p_{20}$, PPS1 sets its probabilities of using strategies lower than p_{20} to zero and shifts them to $p_1 = p_{20}$ because it brings a better payoff. It will help the convergence to the equilibrium we derived in previous chapters.

As for the first-round auction, PPS1 will know that it cannot keep winning in an equilibrium since PPS2 can always bid higher than PPS1. The only way for PPS1 to utilize this information is leading PPS2 to estimate c_1 lower than actual c_1 and to bid lower than actual c_1. In order for that, PPS1 can choose a bid price $p_1 \leq c_1$ at the second-round auction, accepting a negative payoff. However, it isn't likely to alter the equilibrium we derived in previous chapters. When PPS2 loses to PPS1, it will notice that its estimated c_1 is lower than actual c_1. PPS2 can always correct its estimated c_1, adapt its strategy to it, and win in the following first-round auctions.

The second information, $c_2 < p_{20}$, is meaningful to PPS1 when $p_{20} < c_1$. In this case, PPS1 can utilize this information effectively in the first-round auction. Knowing it can always win the first-round auction bidding with the bid price $p_1 \geq p_{20}$, PPS1 sets its probabilities of using strategies higher than p_{20} to zero and shifts them to $p_1 = p_{20}$ because it brings a better payoff. In the same way as the first information, this information helps the dynamics of the repeated auction to converge faster.

As for the second-round auction, PPS1 will know that it cannot keep winning in an equilibrium since PPS2 can always bid lower than PPS1. The only way for PPS1 to utilize this information is leading PPS2 to estimate c_1 higher than actual c_1 and to bid higher than actual c_1. In order for that, PPS1 can choose a bid price $p_1 \geq c_1$ at the first-round auction, accepting a negative payoff. However, it isn't likely to alter the equilibrium we derived in previous chapters. When PPS2 loses to PPS1, it will notice that its estimated c_1 is higher than actual c_1. PPS2 can always correct its estimated c_1, adapt its strategy to it, and win in the following first-round auctions.

Now, we show an example simulation result of replicator analysis when players utilize additional information. Let P_f denote the minimum acceptable price in the first-round auction and P_s denote the maximum acceptable price in the second-round auction. Note that a bidding price equal to P_f or P_s is not acceptable, as in previous chapters. Consider the case $c_1 = 7.0$, $c_2 = 9.0$, $P_f = 5.0$, and $P_s = 11.0$. PPS1 and PPS2 bid both at the first-round and at the second-round auctions, repeatedly. They have sufficient generating capacity to participate in the second-round auctions, but since P_f is lower than their cost, they also try to win in the first-round auctions.

However, even when one of them wins in the first-round auction, it doesn't lower its cost in the following second-round auction, assuming that it has its own demand larger than the amount obtained in the first-round auction.

Payoff matrices are given as follows:

		PPS2's Strategy (p_2)						
		5.5	6.0	6.5	7.0	7.5	8.0	8.5
PPS1's	5.5	(3/4, 7/4)	(0, 3)	(0, 5/2)	(0, 2)	(0, 3/2)	(0, 1)	(0, 1/2)
Strategy	6.0	(1, 0)	(1/2, 3/2)	(0, 5/2)	(0, 2)	(0, 3/2)	(0, 1)	(0, 1/2)
(p_1)	6.5	(1/2, 0)	(1/2, 0)	(1/4, 5/4)	(0, 2)	(0, 3/2)	(0, 1)	(0, 1/2)

Table 7.1: Payoff matrix for the first-round auction.

		PPS1's Strategy (p_1)						
		7.5	8.0	8.5	9.0	9.5	10.0	10.5
PPS2's	9.5	(1/2, 0)	(1, 0)	(3/2, 0)	(2, 0)	(5/4, 1/4)	(0, 1/2)	(0, 1/2)
Strategy	10.0	(1/2, 0)	(1, 0)	(3/2, 0)	(2, 0)	(5/2, 0)	(3/2, 1/2)	(0, 1)
(p_2)	10.5	(1/2, 0)	(1, 0)	(3/2, 0)	(2, 0)	(5/2, 0)	(3, 0)	(7/4, 3/4)

Table 7.2: Payoff matrix for the second-round auction.

From these payoff matrices, we clearly see that $(p_1, p_2) = (6.5, 7.0)$ is the Nash equilibrium of the non-repeated first-round auction and $(p_1, p_2) = (9.0, 9.5)$ is the Nash equilibrium of the non-repeated second-round auction. We can expect that $p_2 = 7.0$ is the equilibrium of the repeated first-round auction and $p_1 = 9.0$ is the equilibrium of the repeated second-round auction.

Suppose we have the following initial condition:

First-round

Strategy	5.5	6.0	6.5	7.0	7.5	8.0	8.5
PPS1	0.1000	0.1000	0.8000	-	-	-	-
PPS2	0.4000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000

Second-round

Strategy	7.5	8.0	8.5	9.0	9.5	10.0	10.5
PPS1	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.4000
PPS2	-	-	-	-	0.8000	0.1000	0.1000

Also, suppose that PPS1 and PPS2 utilize additional information in this section when the other player uses a strategy with a probability higher than 0.1050 and wins or ties in an auction. If we didn't use replicator analysis for this simulation, we could start from zero probability and utilize additional information when a strategy is used with any positive probability. In replicator analysis, however, a strategy with zero probability will never be used in future auctions. Thus, we start from a probability of 0.1000 and utilize additional information when the probability grows higher than 0.1050.

After 15 auctions, the probability distribution is

First-round

Strategy	5.5	6.0	6.5	7.0	7.5	8.0	8.5
PPS1	0.1000	0.1042	0.7958	-	-	-	-
PPS2	0.3282	0.0893	0.1222	0.1415	0.1221	0.1967	0.0000

Second-round

Strategy	7.5	8.0	8.5	9.0	9.5	10.0	10.5
PPS1	0.0000	0.1974	0.1220	0.1414	0.1221	0.0892	0.3279
PPS2	-	-	-	-	0.7959	0.1042	0.0999

In this distribution, PPS2 doesn't use the strategy $p_2 = 8.5$ in the first-round auction. Since PPS1 used the strategy $p_1 = 8.0$ in the second-round auction with a probability higher than 0.1050, PPS2 noticed that c_1 was lower than 8.0 and it could always win with the strategy $p_2 = 8.0$. Therefore, it doesn't use the strategy $p_2 = 8.5$, and $P(p_2 = 8.5)$ is shifted $P(p_2 = 8.0)$, which has the higher payoff than $p_2 = 8.5$. This evolution of strategy will help the convergence to the expected equilibrium of the repeated first-round auction, $p_2 = 7.0$. In the same way, we can see PPS1 doesn't use the strategy $p_1 = 7.5$ in the second-round auction. Note that the initial condition in this case is chosen so that these evolutions take place. Starting from some initial conditions, the players can never utilize additional information.

After 1000 auctions, the probability distribution converges to

First-round

Strategy	5.5	6.0	6.5	7.0	7.5	8.0	8.5
PPS1	0.0855	0.1015	0.8130	-	-	-	-
PPS2	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000

Second-round

Strategy	7.5	8.0	8.5	9.0	9.5	10.0	10.5
PPS1	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
PPS2	-	-	-	-	0.8133	0.1012	0.0855

This result is same as the result when players don't utilize additional information. This gives rise to the following two suggestions. Most importantly, the equilibrium is not affected by the utilization of additional information. The other suggestion is that the advantage in the convergence speed is canceled at least in this case because the evolutions of strategy enabled by the utilization of additional information are not rate-determining.

7.4 Conclusion

In order to analyze the repeated auction, three kinds of analytical approaches — the iterated elimination of dominated strategies, Markov chain analysis, and replicator dynamics analysis — were used in this thesis. All analytical approaches show that in equilibrium, the company with the lower cost wins the auction by bidding at the other company's cost.

Agent-based simulation shows similar results in most cases, but the learning algorithm and the strategy selection have a notable impact on the results. Generally speaking, the average learning algorithm makes it easier to predict the strategy to be used than the other two algorithms, and between the two strategy selections we have studied, the ϵ-greedy selection also makes it easier to predict the strategy to be used than the payoff-weighted selection. Throughout the simulations, tacit collusion was not observed, though it was plausible that tacit collusion could be formed when traders knew the Pareto efficient strategy.

The results in this thesis suggest the potential value of analytical approaches and agent-based simulation to analyze the repeated electricity auction. All these investigations are undertaken on different suppositions. Markov chain analysis assumes that the change of strategies obeys a Markov process, and replicator dynamics analysis assumes that the change of strategies obeys the discrete time replicator equation. Finally, for the agent-based simulation, the specific learning algorithms and strategy selection methods are assumed. Under these different assumptions, these analytical approaches and the agent-based simulations result in similar equilibrium conditions, which suggests the validity of assumptions.

Appendix A

MATLAB Code for the Replicator Analysis

A.1 Replicator Analysis with Two Sellers

```
% Replicator Equations
function [x_1, x_2] = replicator()

alpha = 0.02;

% Initial value of the proportion of players
% using each strategy
% Uniform Distribution
x_1 = [0.2, 0.2, 0.2, 0.2, 0.2]';
x_2 = [0.25, 0.25, 0.25, 0.25]';
% Distribution close to Pareto efficient
% x_1 = [0.0001, 0.0001, 0.0001, 0.0001, 0.9996]';
% x_2 = [0.0001, 0.0001, 0.0001, 0.9997]';

% Payoff matrices
A_1 = [2.0 2.0 2.0 2.0; 1.5 3.0 3.0 3.0; 0.0 2.0 4.0 4.0;
```

```
    0.0 0.0 2.5 5.0; 0.0 0.0 0.0 3.0];
A_2 = [0.0 0.0 0.0 0.0; 0.5 0.0 0.0 0.0; 1.0 1.0 0.0 0.0;
    1.0 2.0 1.5 0.0; 1.0 2.0 3.0 2.0]';

for i = 1:100
    % Each row of p_1 represents PPS1's payoff
    % with respect to each PPS1's strategy
    p_1 = A_1 * x_2;

    % PPS1's average payoff over its strategies
    p_10 = sum(x_1 .* p_1) .* ones(5, 1);

    % Update PPS1's proportion
    x_1 = x_1 + alpha .* x_1 .* (p_1 - p_10);

    % Each column of p_2 represents PPS2's payoff
    % with respect to each PPS2's strategy
    p_2 = A_2 * x_1;

    % PPS2's average payoff over its strategies
    p_20 = sum(x_2 .* p_2) .* ones(4, 1);

    % Update PPS2's proportion
    x_2 = x_2 + alpha .* x_2 .* (p_2 - p_20);
end
```

A.2 Replicator Analysis with Three Sellers

```
% Replicator Equations for Three Sellers
function [x_1, x_2, x_3] = replicator_by3()

alpha = 0.02;

% Initial value of the proportion of players
% using each strategy
% Uniform Distribution
x_1 = [1/6, 1/6, 1/6, 1/6, 1/6, 1/6]';
x_2 = [1/5, 1/5, 1/5, 1/5, 1/5]';
x_3 = [1/4, 1/4, 1/4, 1/4]';
% Distribution close to Pareto efficient
% x_1 = [0.0001, 0.0001, 0.0001, 0.0001, 0.0001, 0.9995]';
% x_2 = [0.0001, 0.0001, 0.0001, 0.0001, 0.9996]';
% x_3 = [0.0001, 0.0001, 0.0001, 0.9997]';

% Payoff matrices
% Payoff for PPS1 (A1_8 is a payoff matrix when p3=8.)
A1_8  = [1.0 1.0 1.0 1.0 1.0; 1.0 2.0 2.0 2.0 2.0; 0.0 1.0 1.5 1.5 1.5;
    0.0 0.0 0.0 0.0 0.0; 0.0 0.0 0.0 0.0 0.0; 0.0 0.0 0.0 0.0 0.0];
A1_9  = [1.0 1.0 1.0 1.0 1.0; 1.0 2.0 2.0 2.0 2.0; 0.0 1.5 3.0 3.0 3.0;
    0.0 0.0 4/3 2.0 2.0; 0.0 0.0 0.0 0.0 0.0; 0.0 0.0 0.0 0.0 0.0];
A1_10 = [1.0 1.0 1.0 1.0 1.0; 1.0 2.0 2.0 2.0 2.0; 0.0 1.5 3.0 3.0 3.0;
    0.0 0.0 2.0 4.0 4.0; 0.0 0.0 0.0 5/3 2.5; 0.0 0.0 0.0 0.0 0.0];
A1_11 = [1.0 1.0 1.0 1.0 1.0; 1.0 2.0 2.0 2.0 2.0; 0.0 1.5 3.0 3.0 3.0;
    0.0 0.0 2.0 4.0 4.0; 0.0 0.0 0.0 2.5 5.0; 0.0 0.0 0.0 0.0 2.0];
A1 = cat(3, A1_8, A1_9, A1_10, A1_11);
% Payoff for PPS2
```

```
A2_8  = [0.0 0.0 0.0 0.0 0.0; 0.5 0.0 0.0 0.0 0.0; 1.0 2/3 0.0 0.0 0.0;
    1.0 1.0 0.0 0.0 0.0; 1.0 1.0 0.0 0.0 0.0; 1.0 1.0 0.0 0.0 0.0];
A2_9  = [0.0 0.0 0.0 0.0 0.0; 0.5 0.0 0.0 0.0 0.0; 1.0 1.0 0.0 0.0 0.0;
    1.0 2.0 1.0 0.0 0.0; 1.0 2.0 1.5 0.0 0.0; 1.0 2.0 1.5 0.0 0.0];
A2_10 = [0.0 0.0 0.0 0.0 0.0; 0.5 0.0 0.0 0.0 0.0; 1.0 1.0 0.0 0.0 0.0;
    1.0 2.0 3/2 0.0 0.0; 1.0 2.0 3.0 4/3 0.0; 1.0 2.0 3.0 2.0 0.0];
A2_11 = [0.0 0.0 0.0 0.0 0.0; 0.5 0.0 0.0 0.0 0.0; 1.0 1.0 0.0 0.0 0.0;
    1.0 2.0 3/2 0.0 0.0; 1.0 2.0 3.0 2.0 0.0; 1.0 2.0 3.0 4.0 5/3];
A2 = cat(3, A2_8, A2_9, A2_10, A2_11);
A2 = permute(A2, [2 3 1]);
% Payoff for PPS3
A3_8  = [0.0 0.0 0.0 0.0 0.0; 0.0 0.0 0.0 0.0 0.0; 0.0 1/3 0.5 0.5 0.5;
    0.0 0.5 1.0 1.0 1.0; 0.0 0.5 1.0 1.0 1.0; 0.0 0.5 1.0 1.0 1.0];
A3_9  = [0.0 0.0 0.0 0.0 0.0; 0.0 0.0 0.0 0.0 0.0; 0.0 0.0 0.0 0.0 0.0;
    0.0 0.0 2/3 1.0 1.0; 0.0 0.0 1.0 2.0 2.0; 0.0 0.0 1.0 2.0 2.0];
A3_10 = [0.0 0.0 0.0 0.0 0.0; 0.0 0.0 0.0 0.0 0.0; 0.0 0.0 0.0 0.0 0.0;
    0.0 0.0 0.0 0.0 0.0; 0.0 0.0 0.0 1.0 1.5; 0.0 0.0 0.0 1.5 3.0];
A3_11 = [0.0 0.0 0.0 0.0 0.0; 0.0 0.0 0.0 0.0 0.0; 0.0 0.0 0.0 0.0 0.0;
    0.0 0.0 0.0 0.0 0.0; 0.0 0.0 0.0 0.0 0.0; 0.0 0.0 0.0 0.0 4/3];
A3 = cat(3, A3_8, A3_9, A3_10, A3_11);
A3 = permute(A3, [3 1 2]);


for i = 1:1500
    A1_weighted = x_3(1) .* A1(:,:,1) + x_3(2) .* A1(:,:,2) ...
        + x_3(3) .* A1(:,:,3) + x_3(4) .* A1(:,:,4);
    % Each row of p_1 represents PPS1's payoff
    % with respect to each PPS1's strategy
    p_1 = A1_weighted * x_2;

    % PPS1's average payoff over its strategies
```

```
    p_10 = sum(x_1 .* p_1) .* ones(6, 1);

    % Update PPS1's proportion
    x_1 = x_1 + alpha .* x_1 .* (p_1 - p_10);

    A2_weighted = x_1(1) .* A2(:,:,1) + x_1(2) .* A2(:,:,2) + x_1(3) .* A2(:,:,3)
        + x_1(4) .* A2(:,:,4) + x_1(5) .* A2(:,:,5) + x_1(6) .* A2(:,:,6);
    % Each column of p_2 represents PPS2's payoff
    % with respect to each PPS2's strategy
    p_2 = A2_weighted * x_3;

    % PPS2's average payoff over its strategies
    p_20 = sum(x_2 .* p_2) .* ones(5, 1);

    % Update PPS2's proportion
    x_2 = x_2 + alpha .* x_2 .* (p_2 - p_20);

    A3_weighted = x_2(1) .* A3(:,:,1) + x_2(2) .* A3(:,:,2) + x_2(3) .* A3(:,:,3)
        + x_2(4) .* A3(:,:,4) + x_2(5) .* A3(:,:,5);
    % Each column of p_3 represents PPS3's payoff
    % with respect to each PPS3's strategy
    p_3 = A3_weighted * x_1;

    % PPS3's average payoff over its strategies
    p_30 = sum(x_3 .* p_3) .* ones(4, 1);

    % Update PPS3's proportion
    x_3 = x_3 + alpha .* x_3 .* (p_3 - p_30);
end
```

Bibliography

[1] R.Rajaraman, F.Alvarado, "Optimal Bidding Strategy in Electricity Markets Under Uncertain Energy and Reserve Prices," *PSERC Working Paper* (2003)

[2] A.Mateo, E.F.Sánchez-Úbeda, A.Muñoz, J.Villar, A.Saiz, J.T.Abarca and E.Losada, "Strategic Bidding under Uncertainty using Genetic Algorithms," *PMAPS2000 : 6th International Conference on Probabilistic Methods Applied to Power Systems. Funchal, Madeira-Portugal*, September 25-28 (2000)

[3] Á.Baíllo, M.Ventosa, M.River and A.Ramos, "Strategic Bidding in a Competitive Electricity Market: A Decomposition Approach," *IEEE Porto PowerTech Porto-Protugal*, September 10-13 (2001)

[4] H.Gintis, "Game Theory Evolving," *Princeton University Press* (2000)

[5] J.Bower, D.Bunn, "Experimental Analysis of the Efficiency of Uniform-Price Versus Discriminatory Auctions in the England and Wales Electricity Market," *Journal of Economic Dynamics and Control*, no.25, pp.561-592 (2001)

[6] D.Bunn, F.Oliveira, "Agent-Based Simulation - An Application to the New Electricity Trading Arrangements of England and Wales," *IEEE Trans. on Evolutionary Computation*, vol.5, no.5, pp.493-503 (2001)

[7] J.F.Nash, "Equilibrium Points in N-Person Games," *Proceedings of the National Academy of Science* (1950)

[8] J.Hofbauer, K.Sigmund, "Evolutionary Games and Population Dynamics," *Cambridge University Press* (1998)

[9] D.McFadzean, L.Tesfatsion, "A C++ Platform for the Evolution of Trade Networks," *ISU Economic Report*, No.39 (1999)

[10] T.D.H.Cau, E.Anderson, "A Co-evolutionary Approach to the Tacit Collusion of Generators in Oligopolistic Electricity Markets: Piecewise Linear Bidding Structure Case," *IEEE Power Engineering Society Summer Meeting*, vol.3, pp.1534-1540 (2002)